University of Alberta

An Advance Reservation Scheduling System

by

Lei Wang

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Computing Science

Edmonton, Alberta
Spring 2004

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-96564-3
Our file Notre référence
ISBN: 0-612-96564-3

The author has granted a non-exclusive license allowing the Library and Archives Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou aturement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

The Internet is a multi-purpose communication network where users and applications with different service demands share network resources. To meet these demands, mechanisms for quality of service in the network are proposed. These mechanisms offer service level agreements to the clients by performing traffic scheduling and network resource reservations. This thesis is focused on resource scheduling and transmission scheme issues associated with resource reservations and on supporting resource reservations in advance.

We present two scheduling algorithms and three transmission schemes for video distribution application in an advance reservation system and try to find out which one or combination is the most efficient. The objective of the scheme is to minimize rejection probability of the customer request and to utilize the bandwidth efficiently. Therefore, a big chunk of data is segmented into smaller pieces in order to make use of the small slot of the bandwidth. Also, reservations from different sources for the same video are aggregated as their paths merge toward the server.

We use simulation to examine the rejection probability of the customer request and the bandwidth utilization for each scheduling algorithm and transmission scheme at difference work load. We also present a comparison of these scheduling algorithms.

To all the people I love and who loves me

Acknowledgements

This thesis is a result of efforts put in by many people. I am greatly indebted to my supervisor, Prof. Janelle Harms and would like to take this opportunity to thank her for providing all the support and guidance throughout the project. I would also like to thank my examining committee Prof. and Prof. for taking interest in the thesis and providing invaluable suggestions.

Contents

1 Dissertation Overview **1**

1.1 Introduction 1

1.1.1 Why QOS (quality of service) 2

1.1.2 Why advance reservation 3

1.2 Problem Statement 6

1.3 Thesis Contribution 7

1.4 Thesis Outline 8

2 Related Work **9**

2.1 Video-On-Demand 9

2.1.1 An Overview of VoD 10

2.1.2 Set-Top Box 12

2.1.3 Video Server 12

2.1.4 Video/Audio Standard 13

2.1.5 Transmission Protocols 13

2.2 Immediate Reservation (ImRe) 17

2.2.1 Resource Reservation Protocol 17

2.2.2 RSVP (Resource ReSerVation Protocol) 18

2.2.3 Differentiated Services (DiffServ) 21
2.3 Advance Reservation 21
2.3.1 General Model for Advance Reservation Protocol . . . 22
2.3.2 An Implementation of ReRA using RSVP 25
2.3.3 The Benefits of Advance Reservation 26
2.4 Summary 28

3 Admission Control and Scheduling Algorithm **29**
3.1 Motivation 29
3.2 Immediate Reservation Model 31
3.3 Advance Reservation Model 33
3.4 Advance Reservation Scheduling Algorithms 34
3.4.1 Earliest Start-time First 34
3.4.2 Latest Start-time First 35
3.5 Advance Reservation Transmission Schemes 36
3.5.1 Extended Scheme 37
3.5.2 Segmented Scheme 38
3.5.3 Concentrated Scheme 42
3.6 Merging 45
3.7 Summary 46

4 Comparison of Proposed Schemes **47**
4.1 Introduction 47
4.2 Simulation Model 47
4.2.1 Parameters and Workload 48
4.2.2 Performance Metrics 49

4.2.3 Simulation Methodology 50
4.3 Reservation Results 52
4.3.1 Advance Reservation Results 53
4.3.2 Mixed Advance and Immediate Reservation 57
4.3.3 Merging 68
4.3.4 Voice and Data Requests 71
4.4 Summary 74

5 Conclusions and Future Work **77**
5.1 Conclusion 77
5.2 Future Work 79

Bibliography **81**

List of Figures

2.1 VoD System 11
2.2 RSVP in Hosts and Routers 19
2.3 Direction of RSVP messages 20
2.4 Immediate Reservation 23
2.5 Advance Reservation 24

3.1 Channels and Time Slots 31
3.2 Logical Channel 32
3.3 Extended Scheme 37
3.4 Segmented Scheme 38
3.5 Concentrated Scheme 39
3.6 Continuous Time Slot for a Video 40
3.7 Time-slot in Concentrated Scheme 43

4.1 Earliest Start-time First: Advance Reservation 54
4.2 ESF-Concentrated: Bandwidth Utilization 55
4.3 ESF-Extended: Bandwidth Utilization 56
4.4 Latest Start-time First: Advance Reservation 57
4.5 Concentrated Scheme: ESF vs. LSF, Advance Reservation . . 58
4.6 Extended Scheme: ESF vs. LSF, Advance Reservation 58

4.7 Segmented Scheme: ESF vs. LSF, Advance Reservation 59
4.8 Earliest Start-time First: 10% ImRe 60
4.9 Latest Start-time First: 10% ImRe 60
4.10 Concentrated Scheme: ESF vs. LSF, 10% ImRe 62
4.11 Extended Scheme: ESF vs. LSF, 10% ImRe 62
4.12 Segmented Scheme: ESF vs. LSF, 10% ImRe 63
4.13 Impact of 10% ImRe 63
4.14 Latest Start-time First: ReRA and ImRe 64
4.15 Earliest Start-time First: ReRA and ImRe 65
4.16 Latest Start-time First: 90% Advance Reservation, Delay . . . 66
4.17 Latest Start-time First: Merging, No Delay 68
4.18 Earliest Start-time First: Merging, No Delay 72
4.19 Latest Start-time First: Bandwidth Utilization 72
4.20 Earliest Start-time First: Bandwidth Utilization 73
4.21 ESF Bandwidth Utilization Comparison 73
4.22 LSF Bandwidth Utilization Comparison 75
4.23 Rejection probability for Video when Voice and Data Requests are Added 75
4.24 Number of Voice Requests Accepted 76

List of Tables

4.1 Confidence Interval: ESF 53

4.2 Rejection Proportion, ReRA vs. ImRe (ESF) 59

4.3 Rejection Proportion, ReRA vs. ImRe (LSF) 61

4.4 Rejection Proportion for ImRe Requests (No delay) 67

4.5 Rejection Proportion for ImRe Requests (Threshold=600) . . 68

4.6 Merging vs. Non-Merging (LSF, NO Delay, Mean Request=60) 69

4.7 Merging vs. Non-Merging (ESF, NO Delay, Mean Request=60) 70

5.1 Performance Comparison: ReRA, Non-Merging 78

5.2 Performance Comparison: ReRA and ImRe, Non-Merging . . 78

5.3 Performance Comparison: ReRA, Merging 78

5.4 Performance Comparison: ReRA and ImRe, Merging 79

Chapter 1

Dissertation Overview

1.1 Introduction

The Internet has become the ubiquitous network for global data communication and may also become the same for the whole of telecommunication. The telecommunication network is largely digital and according to [35], in the United States the data communication on this network is growing at a rate of 100% per year; while the voice traffic is growing 8% per year. Also the processing power of microprocessors will double every 18 months according to Moore's Law, and the storage media becomes cheaper and cheaper at the same time. The cable modem and Asymmetric Digital Subscriber Line (ADSL) are becoming more and more popular in the world. All of these will allow applications to consume data at a much higher bit rate. Then services such as high speed Internet access and video on-demand will all be possible over the channels of the cable TV or telephone network as add-on services.

However this imposes several problems on the network environment primarily related to the provision of support for real-time video and audio transmissions, which require both high bandwidth and end-to-end control of delay

and delay jitter. These Quality of Service (QoS) parameters need to be guaranteed by the network.

To set up services, clients and providers establish service level agreements that give clients different access to resources when there is contention for using them. Providers meet the service demands by scheduling their resources for various purposes. Resources are finite and the objective is to schedule these resources so clients have the best overall network utility. There are cases where resources are not, or cannot be, scheduled to meet further service requests while still meeting already established service level agreements. Therefore, providers will occasionally have to reject some service requests. In this thesis, we define a scheduling and admission architecture where clients can request their network resources immediately as they are needed or ahead of time through advance reservations.

The purpose of this introduction is to raise and motivate the issues that are covered in this thesis. In Section 1.1.1, I motivate why there is a need for quality of service and resource reservations at all. In section 1.1.2, I motivate why there is a need for supporting resource reservation in advance.

1.1.1 Why QOS (quality of service)

Traditionally, the Internet has offered a single level of service known as best-effort service. In a pure best-effort model, the resulting quality of service depends on the current traffic load over the links involved. The only way to provide different service levels is by allocating different bandwidth at the access links. In the rest of the network, there are no mechanisms for providing different service levels. This model was sufficient for a homogeneous user community and a small set of applications that could adapt and do relatively

well under various conditions. The network could be provisioned to deliver an acceptable quality of service to everybody, most of the time.

Now, many applications that have very different requirements on network service quality are running on the Internet. Interactive applications and other real-time applications (e.g., video distribution, Internet telephony, multimedia conferencing etc) require low round-trip delay, low jitter and a minimum throughput to provide customers satisfied services. The users of these applications vary from business users to home users. It is hard to use a single level of service to satisfy different customers. A new model of multiple level of service is needed to support different requirements. The users who want better quality of service will get options to pay more than the users who only need limited service. The network resources will be utilized more efficiently when the users have to pay more to get more resources. It is also a good solution for those users who want more resources and are willing to pay for them.

Resource management provides a way to offer applications reliability with respect to QOS. A resource management system controls the access to scarce system resources needed for audio and video data processing. It checks whether additional service requests can be satisfied, if yes, the required resources are reserved for that application, if not, the request is rejected. The importance of resource management and the need of corresponding techniques are now widely accepted in the research community.

1.1.2 Why advance reservation

Service agreements involving a minimum expected level of service must involve some kind of admission control to make sure that there are sufficient resources in the network to meet the agreements. Admission control gives the provider

an opportunity to reject service to new clients so that quality of service can be given to clients that were already granted admission. Clients, once admitted, have some expectation of the service. In the traditional telephone network, there is only one level of service. Admission control is performed as we initiate a call by dialing a number. Once admitted, clients know that they have sufficient quality for voice communication and that they may hold on to the call as long as they like.

This service model may not be appropriate for the Internet. First, there is a new set of real-time applications involving many users. Carrying out sessions (e.g., conferences, lectures, games, etc) with these applications often require a certain degree of planning so that the participants are attending at the same time. Booking network resources should be part of this planning. It is unfortunate if a planned event has to be cancelled with late notice because it turns out there are insufficient network resources at the time.

Here is an example. Consider, for instance, a meeting room (conferencing) scenario which has to be supported by multimedia systems. Traditionally, a meeting will be scheduled for a specific time in a selected room in order to allow for arrangement of all participants. To be sure that the room will be available at the scheduled time, a reservation is made, in some form of a meeting room calendar, before the meeting starts. The time between the reservation and the meeting itself may vary from short intervals, e.g., hours or a few days, to very long periods, e.g., months. In addition to "one time events", meetings such as department meetings occur periodically. To support these "virtual meeting room" scenarios in a network, the resource reservation system must offer mechanisms to reserve in advance the resources needed during the conference, i.e., routers, and end-system resources.

The resource reservation methods have been divided into two groups: Im-

mediate Reservation(ImRe) and Resource Reservation in Advance (ReRA). In ImRe, users reserve network resources at the time they need them without specifying the duration of the usage. ReRA, on the contrary, reserves the network resources in advance and specifies the time from which the resources will be used as well as the duration of the reservation.

ReRA is not only needed for conferencing but for other scenarios such as video-on-demand as well. This resembles a video rental scenario where a user 'orders' a video for a specific time. It means the resources necessary to retrieve, transfer and present the video in the video-on-demand systems have to be reserved in advance.

ImRe may be able to guarantee the presentation quality of a video, but the user also expects a reasonable response time for receiving the video. Although convenient to the user, this ImRe service can result in delays that are unacceptable or the request can even be rejected. An alternative is to have the users make reservations in advance. Then a request arrives specifying both a desired video and a preferred start time.

Second, there is an increasing demand for long-term planning of trunk network capacity.[14] This is often provided through virtual leased lines and virtual private networks. These services today involve tunnelling and manual configuration of routers along the paths. There is a demand for having automated setup of such services. Trunk bandwidth contracts generally involve planning of network resources over time (e.g., time-limited contracts, various bandwidths at different times of the day etc). It is reasonable that admission requests can include timing information to allow this.

Third, the network service providers themselves obtain additional benefits when (at least some of) their clients book network resource in advance. First,

the service provider can estimate more accurately the demand for network resources. Second, for the advance request, service can run expensive, time-consuming but highly resource-efficient and scalable algorithms for routing, resource reservation and admission control.

1.2 Problem Statement

Allowing the customers to make advance resource reservations should be a part of a communications architecture to provide better service to the customers. Whether advance reservations are actually needed depends on future resource scarcity. Where resources are plentiful, not even immediate reservation may be necessary. But where resources are so scarce that customers may not get the resources they want even with immediate reservation, it will make more sense to reserve the resources in advance.

Researchers have proposed several techniques for supporting advance reservation of network resources that have better performance than immediate reservation [1][2][11]. It is interesting to observe that the basic mechanism employed is remarkably similar in these different proposals. The basic idea is that each node separately manages its resources over the future time by dividing the time domain into time-intervals and dividing bandwidth into channels. A time-interval within a channel is allocated to a request that is delineated by the start and end time. But when they get a request for a channel, e.g. a request for a video stream, they will allocate a big chunk of a channel with a continuous time-interval that equals the length of the video stream.

The problem here is that a channel will be fragmented by the time-intervals in this type of algorithm. A small segment of a free channel can not be allocated if it is shorter than any requested duration. So, a request may be

rejected even though there is still a lot of bandwidth available in the channels. Then the utilization of bandwidth will be low.

In order to decrease the request rejection probability and satisfy most user requests, and increase the utilization of bandwidth, a new request admission and resource scheduling system is needed.

1.3 Thesis Contribution

The aim of our work is to investigate a new scheduling and admission algorithm for advance reservation systems which can accept more requests and utilize the bandwidth better. The contributions of this thesis are listed as follows.

- A novel data stream segmentation scheduling and admission scheme is presented.
- Data pre-fetch is incorporated with our scheduling and admission scheme, which gives us higher throughput.
- Request aggregation is incorporated with our scheduling and admission scheme, which gives us lower request rejection.

We also present the results of a systematic simulation study, which illustrates the performance of our scheduling and admission algorithm in immediate reservation system and advance reservation system. The simulation is also used to compare the performance of immediate reservation and advance reservation services.

Our algorithm is designed for bandwidth consuming applications (eg, storage data migration, video streaming), but it will still get high throughput when massive data and other services (eg. voice) are transmitted together.

1.4 Thesis Outline

This thesis begins with an introduction to the resource reservation system and the goals to be achieved by this thesis. Chapter 2 gives an overview of video-on-demand(VoD) and Resource Reservation in Advance (ReRA) and some existing approaches in detail. In Chapter 3 we present our admission control and scheduling algorithm for on-demand and advance reservation services in VoD. In Chapter 4 the simulation design is discussed and the results of our simulation are analyzed. Finally we conclude in Chapter 5 with some suggestions for future work.

Chapter 2

Related Work

In this chapter, the resource reservation protocols and the current advance reservation models are studied and analyzed. The purpose is to find out how the reservation protocol works and which properties are needed to be specified in a general advance reservation model. Also, since we are interested in developing a high performance scheduling scheme in an advance reservation system, we need to see which properties would result in an efficient utilization of resources. Video on demand (Vod) is the main application of our scheme, it will be discussed in the first section of this chapter. As Resource Reservation in Advance (ReRA) is developed from the resource reservation protocols, we will present the currently used reservation mechanism "Immediate Reservation"(ImRe) in the second section to find out how ReRA is different from ImRe and why it is better than ImRe.

2.1 Video-On-Demand

Video-on-Demand (VoD) is an interactive multimedia entertainment

system. It works just like the cable or satellite television at your home, but the customer does not have to watch the scheduled programs, he can select a video from a large video database and watch that video at his preferred time. The application for VoD also includes interactive learning, catalog shopping and news and weather forecasting. VoD is bringing us tremendous potential commercial value. The Yankee Group reports that VOD revenues are expected to reach $1.98 billion by 2005 year-end[37].

2.1.1 An Overview of VoD

The Architecture: The typical elements of a video-on-demand architecture which is shown in Figure 2.1 include a cluster of video servers with huge memory to cache the most recent requested videos, a storage subsystem to store the compressed video programmes, a communication network to provide multiple simultaneous channels with guaranteed quality of service, and a set-top box for the user to decode, cache and store the video segments.

The Formats: Videos are stored in a database through the video servers in compressed formats. These formats include MPEG-1, MPEG-2, MPEG-4, Quicktime or Real. After the videos are ordered (either via telephone or remote control), they are transmitted over the communication network and decompressed by set-top box at user's home. In current implementations, picture quality equals to standard VHS, DVD[38] or even HDTV.

The Services: The VoD service providers give their customers a whole bunch of services. While watching movies, customers can pause, fast forward, rewind or record the video. The set-top box works like a VCR without a tape. Another service comes with VoD is Personal Video Recording (PVR). It enables the users to store TV programming and other content, and view

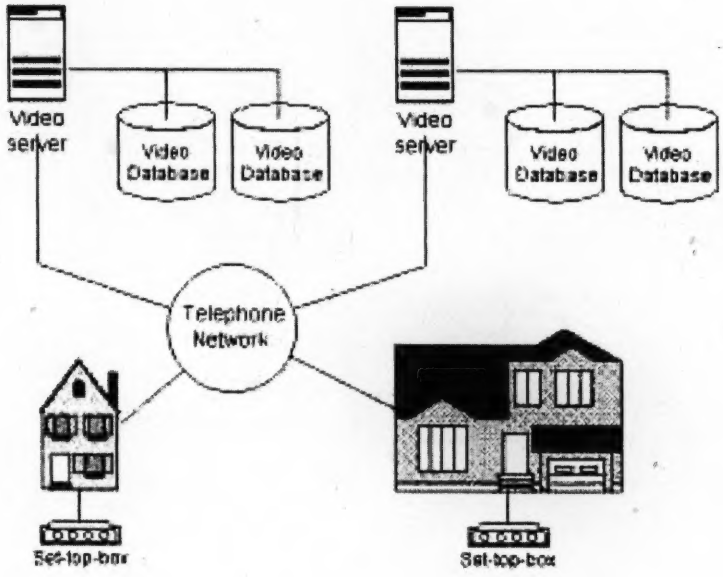

Figure 2.1: VoD System

it anytime. Similar to the case in VOD, PVR users control it as they would control a VCR, and can erase programs at will when no longer of interest. Content can either be stored in the set-top box or on video servers' storage.

The Network: Video-on-Demand is running on the cable company's networks currently. However, the telephone network is the most widespread network. But the problem is that the speed of telephone network is much slower than cable network. A lot of work has been done to improve the capacity of the telephone network and various networking standards and protocols have been developed for multimedia transmission. We will describe then later in this chapter.

Now we will have a closer look at the set-top-box, video server, media compression standards and transmission protocols.

2.1.2 Set-Top Box

A Set-top box is the device to enable the television set to become the user interface for the Internet and to enable the television set to receive and decode the digital video stream from cable broadcast.

A Set-top box is actually a specialized computer containing single or multiple microprocessors and RAM, running operating systems, such as VxWorks or Windows CE. It also includes MPEG decoding chips and other video/audio processing chips.

To the user, a Set-top box is also a VCR-like device. Users can pause, fast-forward, rewind or slow-motion play by using the remote control.

2.1.3 Video Server

VoD is generally built on the client-server architecture where a video server stores compressed video for delivery to the clients on the network. The client will have the set-top box we mentioned in last section. The video server should have the capability to deal with hundreds of real time simultaneous requests from the clients and allocate enough bandwidth for each client. The admission control and scheduling also take place on the video server. In other words, the video server decides if a request has been approved and when the requested the video will be transmitted.

Usually, a service provider has a cluster of video servers, and it should be easily scaled to meet more user requests. The video server also has access to a huge storage pool if media and the storage pool might be up to Terabyte or even Petabyte. A storage area network (SAN) is currently being used for the video server to get fast access to the storage.

The cost of the video server and storage system is a big concern for the service provider. A dynamic media allocation method must be considered. It means that the popular movies could be stored on disks, however, other movies could be put on tapes or CDROMs.

2.1.4 Video/Audio Standard

The multimedia content must be compressed before it can be transmitted over the Internet. The most popular compression technology is from the Motion Picture Experts Group(MPEG)[40]. MPEG-1 defines a framework for coding moving video and audio, significantly reducing the amount of storage with minimal perceived difference in quality. It is aimed at full screen video stored on a CD-ROM and has since been incorporated into the Video CD specification and is used on CD-ROMs. The typical bit rate for MPEG-1 is 1.5Mb/sec. MPEG-2 builds on the MPEG-1 specification, adding further pixel resolutions, support for interlace picture, better error recovery possibilities, more chrominance information formats, non-linear macroblock quantization and the possibility of higher resolution DC components. It is intended for digital television applications and is used for DVD-Video. The typical bit rate for MPEG-2 is 4Mb/sec. The MPEG-4 standard initially differed by its focus on low bit rate coding applications over IP connections. Later its scope was increased to cover a wider range of multimedia applications including videophone, interactive TV, and streaming video. The typical bit rate for MPEG-4 is 64Kb/sec.

2.1.5 Transmission Protocols

We will review some common transmission protocols that are appropriate for multimedia transmission in this section. A typical VOD network can be di-

vided into two levels: backbone and local distribution networks. The backbone connects servers and routers/access nodes together, while the local distribution network links a client site to its nearby router/access node. Each local distribution network link needs a bandwidth of at least 1.5 Mbps - the bandwidth of a single MPEG-1 video stream. The bandwidth of the backbone is usually on the order of hundreds megabits per second, depending on the number of simultaneous connections the VOD network has to support.

Backbone Network

High-speed and low-jitter networks are required for VoD to support hundreds of concurrent connections. The required bandwidth of the backbone is on the order of gigabits per seconds. Three attractive solutions are SONET, Gigabit Ethernet and DWDM.

SONET (Synchronous Optic NETwork)

SONET is a physical layer network technology for optical transmission. It is designed to carry large volumes of traffic over long distances. SONET defines clear interoperability standards to support different vendors' products and to carry different higher layer protocols. With the implementation of SONET, the networks from the carriers all over of world can interconnect with each other.

The SONET standard defines a hierarchy of interface rates that allow data streams at different rates to be multiplexed. The entire bandwidth of a link is devoted to a single channel. The nodes on the channel transmit data at different time slots. SONET establishes Optical Carrier (OC) levels from OC-1 (51.8 Mbps) to OC-192 (9.952 Gbps).

Gigabit Ethernet

Since two individuals at Xerox – Bob Metcalfe and D.R. Boggs – developed Ethernet beginning in 1972, Ethernet has been the dominant networking protocol for three decades. Of all current networking protocols, Ethernet provides the greatest cost performance relative to Token Ring, FDDI and ATM. Gigabit Ethernet builds on top of the Ethernet protocol, but increases speed ten times over Fast Ethernet to 1000Mbps, or 1Gbps.

Gigabit Ethernet was officially adopted on June 25, 1998 as IEEE 802.3z. The IEEE 802.3z standard includes the Gigabit Ethernet MAC, as well as four physical layers that use the 8B/10B encoding originally developed as part of the ANSI Fibre Channel technology.

WDM (Wavelength Division Multiplexing)

Increasing the effective capacity of existing fiber can be accomplished in two ways: increase the bit rate of the existing system and increase the number of wavelength in the fiber. TDM (Time-Divison Multiplexing) is the way to increase the bit rate. SONET is one of the implementation of TDM.

WDM use a completely different method from TDM to increases the carrying capacity of the fiber. In WDM, many wavelengths or colors are combined onto a single fiber. It can simultaneously multiplex signals of 2.5 to 40 Gbps on a single fiber. The effective capacity of existing fiber plant can be increased by a factor of 16 or 32 easily without laying a new fiber.

Like TDM, the resulting capacity is an aggregate of the input signals, but WDM carries each input signal independently of the others. This means that each channel has its own dedicated bandwidth. All signals carried by different wavelength arrive at the same time, rather than being broken up and carried in time slots[6].

Access Networks

Since the MPEG video is running at 1.5 Mbps to 6 Mbps, the local distribution link requires a bandwidth of 1.5 to 6 Mbps. Two kinds of access networks which can provide these kinds of rate are described below.

ADSL (Asymmetric Digital Subscriber Line)

ADSL is a technology for transmitting data at a high bandwidth on existing phone lines to end users. Unlike regular dial-up phone service, ADSL provides continuously network connection. The download stream to the user is much faster than the upload stream. That is why it is called asymmetric line. The speed of downstream is usually from 512 Kbps to 8 Mbps.

ADSL was specifically designed to exploit the one-way nature of most multimedia communication (especially VoD) in which large amounts of information flows toward the user and only a small amount of interactive control information goes back to the service provider. Several experiments with ADSL to real users began in 1996. In 1998, wide-scale installations began in several parts of the U.S. In 2000 and beyond, ADSL has become generally available in urban areas[7].

HFC (Hybrid Fiber Coax)

HFC is replacing normal Cable TV (CATV) lines with lines capable of supporting higher bandwidth and upload stream. HFC is currently being installed by cable TV companies. It migrates the all-coaxial cable deployment to a network with fiber and coaxial cable. Variations on HFC exist and have

been implemented. They include fiber-to-the-feeder, fiber-to-the-hub, fiber-to-the-zone, fiber-to-the-curb, and fiber-to-the-tap. Fiber-to-the-feeder and fiber-to-the-hub are networks with fiber trunks, coaxial distribution links and subscriber drops. The others are networks consisting of fiber trunks and distribution links, but coaxial subscriber drops[5].

To migrate to HFC, the current 300-450 MHz coaxial cables must be replaced by new 750 MHz coax cables. Bi-directional communications are implemented using split-band systems. In North America, the frequency band between 5-54 MHz is used for upstream. The remaining bandwidth is used as downstream. Since clients share the same physical medium, collisions occur when they try to access the medium at the same time. The problem of sharing the transmission medium limits the number of clients attached to a coax tree structure[4].

2.2 Immediate Reservation (ImRe)

2.2.1 Resource Reservation Protocol

Traditional data networks provide only best-effort delivery service, it is impossible for the networks to guarantee constant delay for an application due to variable queueing delays and networks congestion. New components are needed in the network architecture to solve this problem. A lot research has been done recently in developing an integrated services (IntServ) network architecture to support multimedia transmission like video-on-demand, video conference and video games. IntServ is one way for the networks to provide Quality of Service (QoS) which is needed by the applications. The IntServ architecture requires traffic control mechanisms and a resource reservation protocol. The traffic control mechanisms include an admission control algo-

rithm, a packet classifier and a packet scheduler. The admission control decides whether to accept a QoS request based on available resources; the packet classifier recognizes the flow to which locally defined class it belongs to. Packet scheduler uses queue management and other mechanisms to schedule incoming data for transmission based on their service class.

The reservation protocol is used to allocate and release network resources along the data path to ensure QoS. The network will get huge performance benefit from the reservation protocol. Existing reservation protocols such RSVP (Resource ReSerVation Protocol)[15] implement so-called Immediate Reservation (ImRe) where network resources are requested and granted just when it is actually needed and last an unspecified time.

2.2.2 RSVP (Resource ReSerVation Protocol)

RSVP is one part of the enhancement for the current IntServ with support for Quality of Service flows. Hosts use RSVP to request a specific quality of service from the network for particular application data streams. The routers on the data path use RSVP to propagate QoS requests along the route. The routers also keep the RSVP states up to date to provide the requested service. Generally, after the RSVP request from the host is accepted by the networks, the resources in each node on the data path will be reserved.

The RSVP model is shown in Figure 2.2. For a host to make a resource reservation at a router, the RSVP daemon on the router talks with the admission control module and the policy control module locally. The admission control module determines whether the router has sufficient resources in order to provide the requested QoS. The policy control module determines whether the host has got permission to make the reservation. If either one of the checks

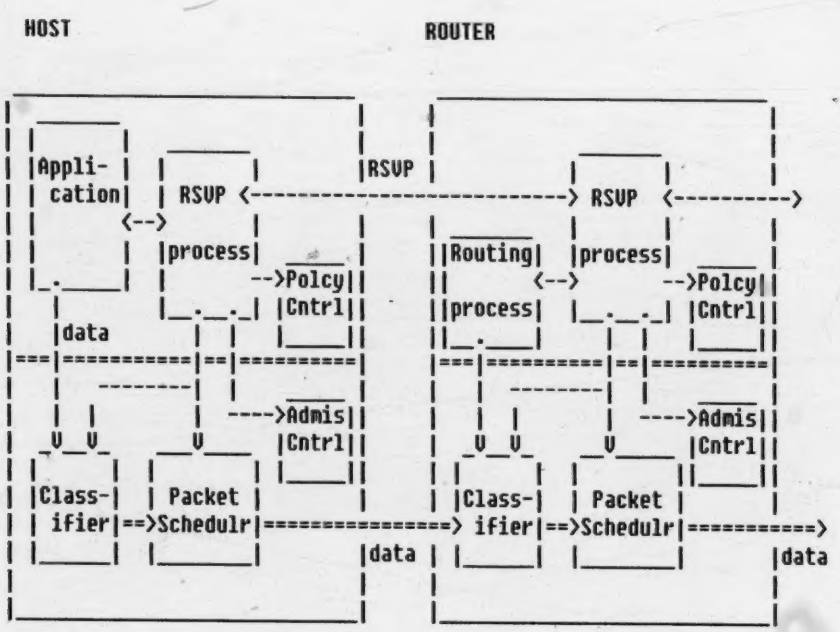

Figure 2.2: RSVP in Hosts and Routers

fails, the RSVP daemon returns an error notification to the application that originated the request. If both checks succeed, the RSVP daemon sets parameters in a packet classifier and packet scheduler to obtain the desired QoS. The packet classifier determines the QoS class for each packet and the scheduler orders packet transmission to achieve the promised QoS for each stream.

RSVP uses the combination of destination address, destination port number and transport layer protocol type to identify a data stream. RSVP is a signaling protocol that carries the bandwidth reservation along a data path that is predetermined by the underneath routing protocol. Figure 2.3 shows an example of RSVP for a multicast session involving one sender S, four routers and three receivers, RCV1, RCV2 and RCV3.

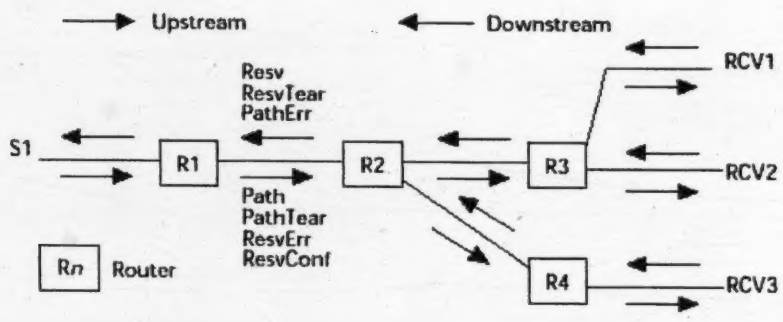

Figure 2.3: Direction of RSVP messages

Path messages and Resv messages are the primary messages used by RSVP. The Path message is initialized from the sender. It carries the characteristics of the sender to the receiver. Routers on the path note the direction of the PATH message and set up reverse path to sender. The traffic receiver sends the Resv message to the traffic sender following the reverse path. It carries the reservation request along the data path between the sender and receiver.

When a reservation is rejected, the receiver will receive an immediate response telling the receiver that the network cannot currently support the request bandwidth or service level. The receiver decides whether to ask the sender to send the data now using best-effort delivery or to wait and repeat the request later.

RSVP is a soft-state protocol, the reservation states must be periodically refreshed. The reservation state is cached in each router. If the underneath network routing protocol alters the data path, RSVP will automatically set up the reservation state along the new data path.

2.2.3 Differentiated Services (DiffServ)

One of the most widely held beliefs about RSVP is that it suffers from scalability problems. RSVP was originally designed to support resource reservation for individual data flows. To avoid the need to maintain per-flow reservation state inside large IP networks, Differentiated services, or DiffServ, is introduced by IETF. The DiffServ divides traffic into a small number of classes and allocates resources on a per-class basis.

Because DiffServ has only a few different classes of traffic, a packet's class can be carried directly within the IP packet header. The field is called Differentiated Services Code Point (DSCP). DSCP is a 6-bit field, which was formerly part of the Type of Service (ToS) byte. In theory we could have 64 different classes of packets. But in practice the number of DSCPs is much smaller[9].

The DSCP tells the router how to treat the the packet from the QoS perspective. The DSCP identifies a "per-hop behavior" or PHB. There are some standard PHBs, and it is also possible to define local PHBs within a network. The standard PHBs include "Default forwarding" where packets do not have any specific agreement; "Expedited forwarding (EF)" where departure rate of an aggregate's packets meets the configured rate; "Assured forwarding (AF)" where four independent forwarding classes, three different drop precedences in each class are defined[9].

2.3 Advance Reservation

In this section, a general model of ReRA is described in comparison with the ImRe; an implementation of setting up ReRA using RSVP is presented and

the benefits of ReRA are introduced.

2.3.1 General Model for Advance Reservation Protocol

Two key factors[22] are used to distinguish ReRA from ImRe protocols:

1. whether the resources are exploited at the reservation time
2. whether the duration of the reservation is indicated at the reservation time

ImRe systems (e.g. RSVP) assume that the resources are immediately used after they have been successfully booked and the duration of the reservations is not specified. A ReRA system, in contrast, is characterized by deferred usage and a known duration.

Several ReRA protocols have been proposed (e.g. in [17] and [22]). The differences between ImRe and ReRA protocol are shown in Figure 2.4 and Figure 2.5.

In the first part of the ReRA protocol, the client issues a REQUEST at time $t_{request}$ and it specifies the nature of its request by indicating how much of the resource capacities will have to be reserved, i.e., it gives a workload specification. It also specifies the start-time t_{start} and duration d_{res} of the reservation. If the request passes the admission control successfully, the service provider will send a CONFIRM to client at time $t_{confirm}$. As a part of the confirmation, each reservation is assigned an identifier for later access.

After setup a reservation successfully, the client may want to re-negotiate the resource requirement. Sometimes the client wants to cancel the reservation, then a RELEASE message will be sent. Sometimes the client wants to increase

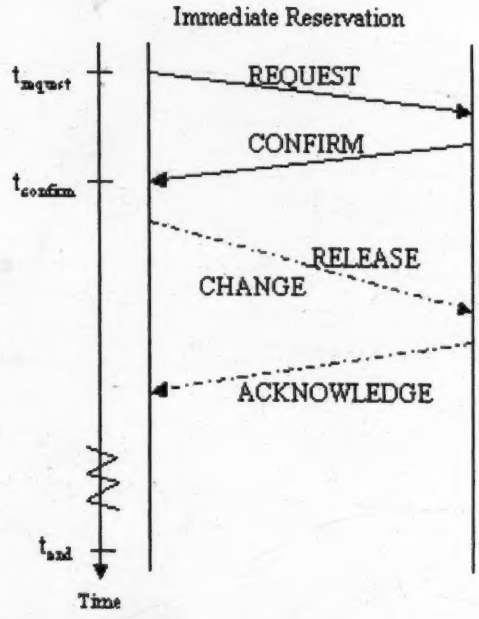

Figure 2.4: Immediate Reservation

or decrease the requested resources, then a CHANGE message will be sent to the service provider. If the service provider accepts the changes or releases the reservation successfully, it will send an ACKNOWLEDGE to the user. Re-negotiation concerns not only the QoS parameters but also the start-time and duration. A re-negotiation can be performed until time t_{end}[17].

If a reservation can not be accepted, the service provider may simply send back a rejection to the user. To make it more flexible, some suggestions can be carried with the rejection:

1. If the duration of a reservation is longer than what the service provider

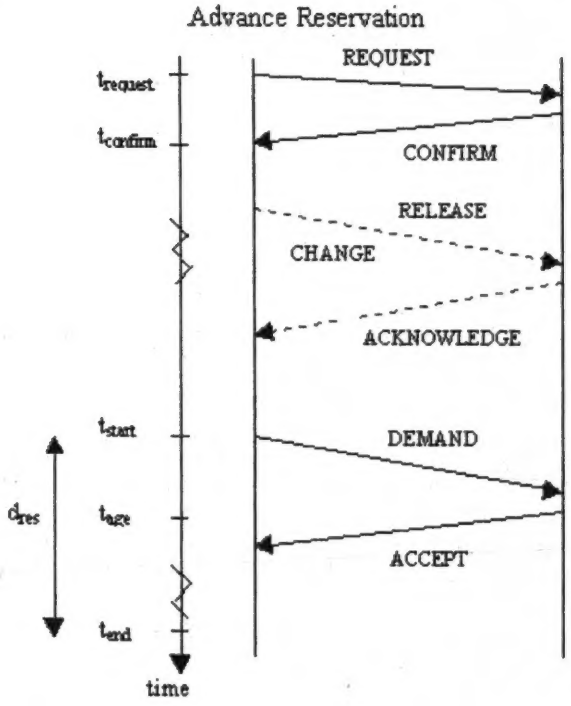

Figure 2.5: Advance Reservation

can support, the service provider could offer a shorter duration.

2. The service provider could suggest a possible start-time when the requested resources are available.

3. The service provider could provide the currently available resources for the request.

The second phase begins shortly before the user intents to exploit its reservation. The client sends a DEMAND to the service provider within the time interval $\{t_{start}, t_{age}\}$ to confirm he will exploit the resource he reserved. If the

sender does not send a DEMAND by then, the reservation will be aged and all the state information corresponding to that reservation will be removed when t_{age} is reached[17]. The client may be requested to show some form of identification in the DEMAND, which the service provider will accept. After receiving the acceptance, the client begin to exploit the reservation.

Normally, the distance between $t_{request}$ and $t_{confirm}$ is very short in a ReRA system. In contrast, t_{start} and $t_{request}$ are possibly quite distant. Let us call I *the notice interval between resource reservation and exploitation:*

$$I = t_{start} - t_{request} \tag{2.1}$$

If I is too small, the advance reservation will be pointless. So in that circumstance ImRe will be used. In a ReRA system, a specific I would be given, say I_{min}. The requests whose notice intervals are less than I_{min} will be rejected or treated as ImRe requests. For a similar reason, an I_{max} for I should be specified to prevent an application to request a reservation too early. This can reduce the number of states which must be kept in the servers or routers.

2.3.2 An Implementation of ReRA using RSVP

ReRA has been implemented in different ways. But quite a few approaches evolve from RSVP. To upgrade RSVP to support ReRA is very attractive since RSVP is widely used on the Internet and only a minor change is required for RSVP. A typical implementation for setting up ReRA with RSVP have been proposed in [17].

To support ReRA, the start-time and duration should be included in the flow specification carried by the RSVP Path and Resv message. The bandwidth from the point of start-time and with required duration will be reserved.

In [17], which is based on RSVP APIs, a new RSVP class is defined to carry ReRA information. This implies that no new message has to be defined in RSVP. Because the ReRA object is an additional and optional object in an RSVP message, the mechanisms for sending and receiving RSVP messages need not to be changed and the semantics of the messages will be preserved for ImRe[17].

As we mentioned in Section 2.2.2, RSVP soft states are created and carried by Path and Resv messages. For reservations made in advance, there is potentially a very large number of Path and Resv messages which must be sent before the client starts to use the bandwidth. To reduce the overhead, we should start sending these messages at a low frequency and increase it when the start time is coming[21].

2.3.3 The Benefits of Advance Reservation

At first glance, supporting ReRA requires the service provider to maintain more information and to use more mechanisms for the ReRA traffic. Will the clients and service provider really benefit from ReRA to balance the complexity it brings? We enumerate some of the benefits in this section.

For the clients

The clients get at least three significant benefits by booking resources in advance[30].

1. *Better Service Quality: They do not need to wait anxiously until the start time, wondering if they can get the required bandwidth. If the service provider can not give the clients the desired resources, the clients*

can change their request or go to another service provider.

2. *Low rejection probability: In most systems, the service provider serves the requests by FCFS (First Come First Service) until the system runs out of resources. If multiple clients request resources for the same time interval, the advance request will naturally get higher priority than immediate request. Therefore with advance reservation, the clients reduce the probability that their requests will be rejected.*

3. *Low cost: The service provider can significantly improve its quality of service, e.g. routing, resource allocation and management, if it can get sufficient advance notice from its clients. The service provider may pass some of these savings to its clients who book resources in advance.*

4. *Pre-fetch: Another advantage of this scheme is the ability to pre-fetch videos for requests that have been scheduled earlier.*

For the service provider

Although the service providers need to provide more support for advance reservation, it is interesting to see that the network service providers themselves get additional benefits when their clients book network resources in advance, these benefits are[30]:

1. *Better service provision and demand forecasting: With advance reservation, the service provider can estimate more accurately the demand for network resources. The service provider can then use this information to manage its resources and schedule its transmission.*

2. *Scalability: The service provider can divide the requests into two categories: advance requests and immediate requests. For advance requests,*

the service provider can use expensive, time-consuming but highly resource-efficient and scalable algorithms for routing and resources management and admission control; for immediate requests, the service provider can use fast algorithms which need not be very scalable.

3. *Better routing: With advance reservation, the network services can significantly improve the routing quality, which is based on these two factors:*

   (a) *The routing system can spend more time on route computation, using more expensive routing algorithms to get better route.*

   (b) *The routing system can also reroute advance flows before data transmission begins. This is a very useful property which will not cause any disruption in service. Such routing capability makes the routing problem off-line, and thus the network can use high performance off-line routing algorithms.*

2.4 Summary

Some related work is presented in this chapter. We gave an overview of Video on Demand system here because it is the major application of the ReRA protocol and it is used as the application in our simulation. As the signalling of IntServ, RSVP is studied in some level of details. To solve the scalability problem of RSVP, DiffServ is introduced by IETF. For a better understanding of ReRA models, we studied the properties of a general resource reservation protocols first, then gave an example of the ReRA model. Both customers and service providers will get benefits from the ReRA protocols.

Chapter 3

Admission Control and Scheduling Algorithm

In the previous chapter, we introduced the resource reservation protocols, the model of advance reservation and the benefits of advance reservation as well. From that we can see advance reservation gives us much room to improve the performance of multimedia transmission on the Internet. In this chapter we propose several admission control and scheduling schemes for advance reservation. In the mean time, some of these schemes can be used in immediate reservation.

3.1 Motivation

Various researchers have proposed techniques for supporting advance reservation of network resources to get better performance than immediate reservation. It is interesting to observe that the basic mechanism employed is remarkably similar in these different proposals. The basic idea is that each node separately manages its resources over the future time by dividing the time domain into time-intervals and dividing the bandwidth of a link into channels.

A time-interval within a channel is allocated to a request that is delineated by the start and end time, and a time-interval is divided into several time slots. But in all of these systems, bandwidth is allocated in a number of continuous time slots.

A channel may become fragmented particularly if only continuous time slots are used for each of the customer requests in those systems. A small segment of a free channel can not be allocated if it is shorter than the duration of any video stream which is currently requested. So, a request may be rejected even though there is still a lot of bandwidth available in the link. Then the bandwidth utilization will be very low. This is shown in Figure 3.1. When a request for a video arrives at time t_1, the start time requested is t_2, and the duration is longer than one time slot, then even though slot A, B, and C are available, they can not be allocated to the video stream, because they are so short that no video stream can fit in it.

Since we are using an advance reservation system, pre-fetch is possible and is a key feature in our system. From Figure 3.1 you can see slot A, B and C are before the requested start time t_2, so if we divide the video stream into small segments where each segment is the length of a time slot, the slot A, B and C can be utilized.

In the following sections, we will give the details of our algorithms. Before that, the basic advance reservation model and some considerations for the implementation are presented.

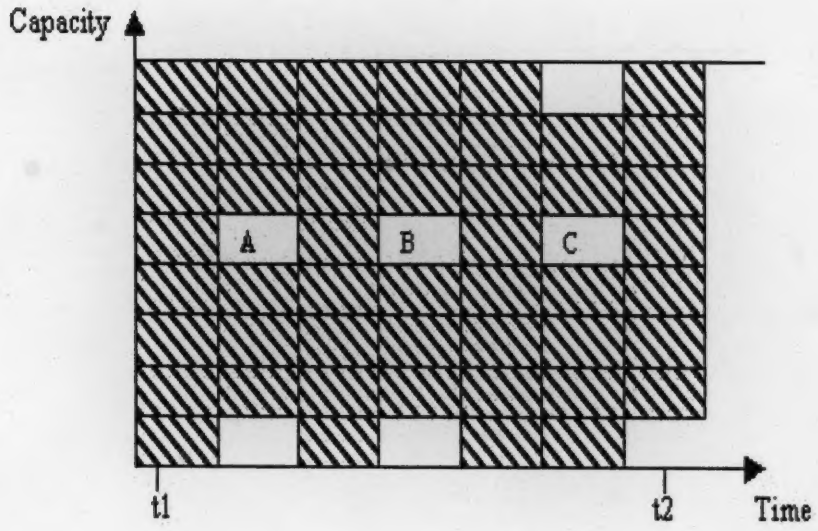

Figure 3.1: Channels and Time Slots

3.2 Immediate Reservation Model

For the immediate reservation service, a request arrives to the server asking for immediate delivery of a video. In other words, the request arrival time is equal to the preferred video start time. The processing time and the time to fetch the video from storage is not explicitly modelled since our focus is on the bandwidth requirements (that is, the number of channels used). This assumption results in smaller delays for the ImRe traffic than would occur in reality. The duration of the reservation is the length of the video which is ordered by the customer.

The starting time of a video or conference and its duration is determined by the user. In general, limiting the possible start times can reduce overhead. The admission control system keeps a calendar for the resources. Fewer possible

start times means you do not need to keep track of very small time intervals in the calendar. This may be reasonable to the user as well. For example, a request at 11:02:00am makes more sense than a request at 11:02:34am.

When a request for a video is coming, if no bandwidth is available, the request will be rejected. Otherwise, the request is assigned to a logical channel. A logical channel is a series of continuous time slots, shown in Figure 3.2. Time slots 1 to 7 make a logical channel. It is very important to clarify that in the pure on demand reservation scenario, the assigned time slots will always be allocated in a physical channel, but in a pure advance reservation or mixed scenario, the assigned time slots may come from different physical channels to form a logical channel except the extended scheme we will present later in which the continuous time slots will make of single channel to simplify the decision making and implementation.

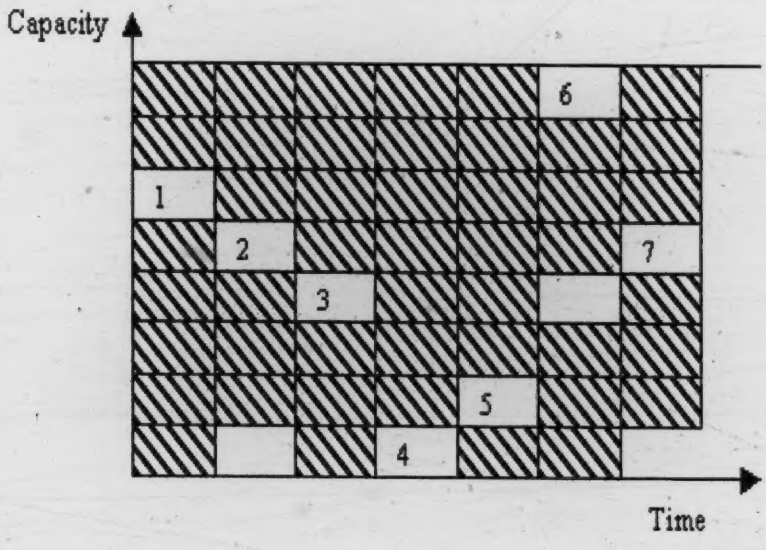

Figure 3.2: Logical Channel

In some circumstances, a user can wait for a video for a short period of time, but may tire of waiting for a period of time which has unpredict length. So we set a threshold for the delay, if the request can not be satisfied within the threshold, it will be rejected. Then the users will not have an endless wait. If they can not get the resources they need before the threshold, they may change to another service provider.

3.3 Advance Reservation Model

Since the load characteristics (duration) of video requests are known as soon as a video choice is made, the requests can be reserved in advance and guaranteed service. For reservation traffic, the request specifies a preferred start time as well as a video. The start time is characterized by the notice interval, the time between arrival and preferred start time. A request arrives to a reservation scheduler; this scheduler decides how to allocate a channel to the request.

In the advance reservation model, bandwidth of the service provider consists of channels. Channels can be allocated to various services. In particular, in our model, N channels are dedicated to a video server or a cluster of servers. The users who share the physical link can access any of the channels. A particular channel is indistinguishable from another from the scheduling point of view. A separate bi-directional link (or channel) is used for communicating requests and acknowledgements between the users and server. Time on each channel is divided into fixed-length units called slots. The duration of a video is measured in slots and the videos are always transmitted at the beginning of a time slot.

3.4 Advance Reservation Scheduling Algorithms

Our system consists two major components: the scheduling subsystem and the transmission subsystem. The scheduling subsystem decides when to schedule requests (early or just in time); the transmission subsystem decides the form of transmission (e.g. in a single contiguous channel or in split up in pieces. We will introduce two scheduling algorithms first in this section and three transmission schemes in the following section.

3.4.1 Earliest Start-time First

Since the requests arrive before the start time, the service provider can begin transmitting the video before the requested start time. If we transmit the video at the earliest possible time, the algorithm will be called Earliest Start-time First (ESF). This requires the user to store the partial or whole video before playback.

In the ESF algorithm, suppose t_{arrive} is the request arrival time, t_{start} is the anticipated start time, d is the duration of the requested video. d is represented by the number of time slots the video will take. The program checks the available bandwidth from t_{arrive} until t_{start}, if the video can fit in a channel, the request will be accepted; otherwise it will be rejected.

The following is the algorithm for our approach.

ESF Algorithm

```
Found=false; i=t_arrive
while ((i <= t_start) and not Found) do
   if d continuous slots beginning with i are available
```

```
        Found=true
    else
        i=i+1
if Found
    accept the request
    this video is scheduled to start to transmit at time i
else
    reject the request
```

3.4.2 Latest Start-time First

Alternatively, the service providers may try to transmit the video as late as possible but still arriving by the start time. This algorithm is called Latest Start time First (LSF).

In this algorithm, we also suppose t_{arrive} is the request arrival time, t_{start} is the anticipated start time, d is the duration of the requested video. The program checks the available bandwidth from t_{start}, if the video can fit in a channel, the request will be accepted; otherwise the program will try to find an available channel starting from the time one slot ahead of t_{start}, and so on, until t_{arrive} is reached. If there is still no channel available, the request will be rejected.

The following is the algorithm for our approach.

LSF Algorithm

```
Found=false; i=t_start
while ((i >= t_arrive) and not Found) do
    if d continuous slots beginning with i are available
```

```
            Found=true;
        else
            i=i-1
    if Found
        accept the request
        this video is scheduled to start to transmit at i
    else
        reject the request
```

3.5 Advance Reservation Transmission Schemes

The previous two algorithm indicate when to schedule; in this section we consider how to transmit. The traditional scheme (Figure 3.3) used to transmit a video uses one continuous channel, which carries the whole video. We name that the "Extended" scheme. In this thesis we propose a new scheme, named the "Segmented" scheme (Figure 3.4). One video stream is divided into segments, each segment has the same length as a time slot. If all of the segments can be allocated in time, the request will be accepted. Since we assume the client has enough buffers to store the entire video before playback, and each segment has an ID, these segments can be delivered discontinuously, and assembled at the user end. We assume that clients have sufficient bandwidth, so these segments can be transmitted in different channels simultaneously. Segments need not be transmitted contiguously, this could utilize fragmented bandwidth more efficiently than a long stream. Another extreme scheme is to not divide bandwidth into channels. In this case, the whole bandwidth is used to transmit a video (Figure 3.5). We call this the "Concentrated" scheme. The "concentrated" scheme may not be very reasonable because the client may not have as high bandwidth as the server. Therefore it is introduced for

comparison purposes only.

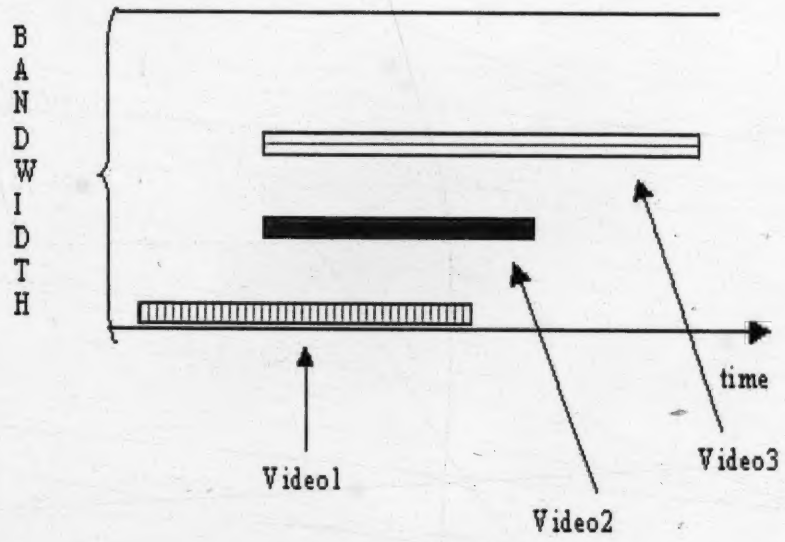

Figure 3.3: Extended Scheme

In the following sections, we upgrade the basic ESF and LSF models to support segmented, concentrated and extended schemes. They are designated as ESF_C (ESF, concentrated), ESF_E (ESF, extended), ESF_S (ESF, segmented), LSF_C (LSF, concentrated), LSF_E (LSF, extended) and LSF_S (LSF, segmented).

3.5.1 Extended Scheme

The algorithm for the extended scheme is almost the same as the basic ESF and LSF algorithms. We use a two dimension array to represent the bandwidth. The rows are channels and the columns are time slots. The duration, d, of a video is represented by the number of time slots it takes. Our task is to find d contiguous columns in the array, where each of these columns has at least one

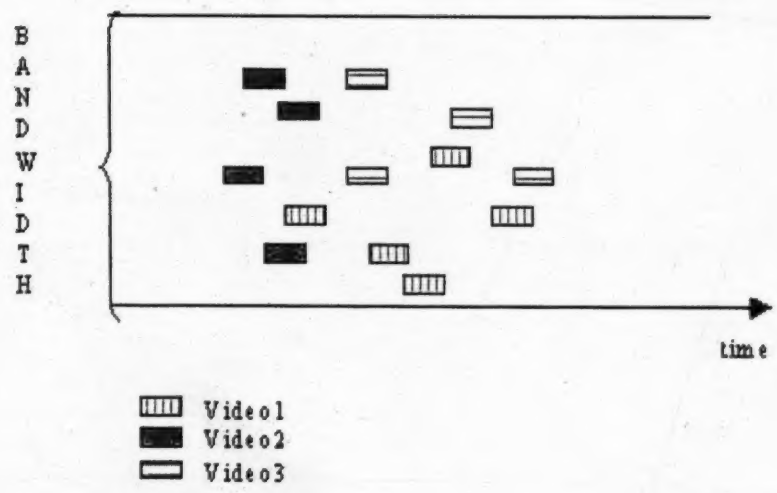

Figure 3.4: Segmented Scheme

empty cell. Figure 3.6 shows the video stream which contains 7 contiguous time slots, although they are not in the same row. The scheduler does not need to know the specific channel for the video, it just needs to know the number of channels used in a time slot and to adjust the number after a request is accepted. Shortly before transmitting the video, a particular physical channel can always be assigned.

3.5.2 Segmented Scheme

The data structure we used in the segmented scheme is the same as that used in the extended scheme. A video is divided into d segments, each segment has the same length of a time slot. The new feature of the segmented scheme is that it does not require d segments in d contiguous columns.

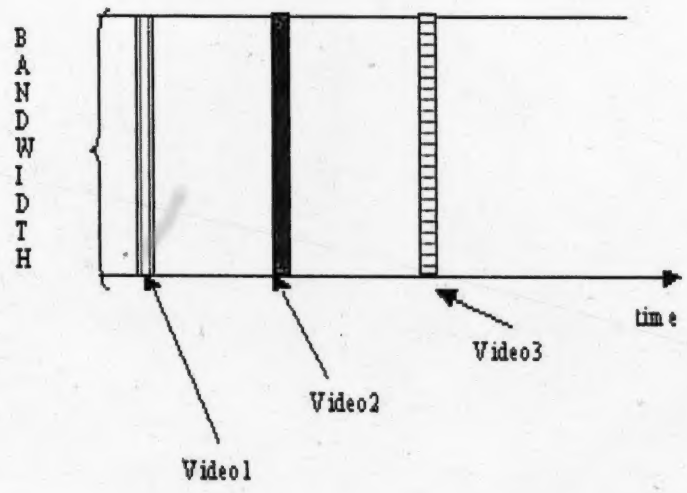

Figure 3.5: Concentrated Scheme

ESF_S

The ESF_S evolves from ESF_E. Each segment can be treated as an independent video stream, and they are numbered from 1 to d. Then they can be assembled easily on the client end. The arrival time and duration for all the segments are the same, the difference among them is that they have different desired start times. The desired time for the first segment is the desired start time of the whole video, t_{start}, the desired start time for the second segment is t_{start} plus the length of a time slot, the desired start time for the third segment is t_{start} plus double the length of a time slot, and so on. If all segments of a video can be allocated between its arrival time and desired start time, the request will be accepted, otherwise it will be rejected.

Let d_trans as the total transmission time for a video. Here is the algorithm

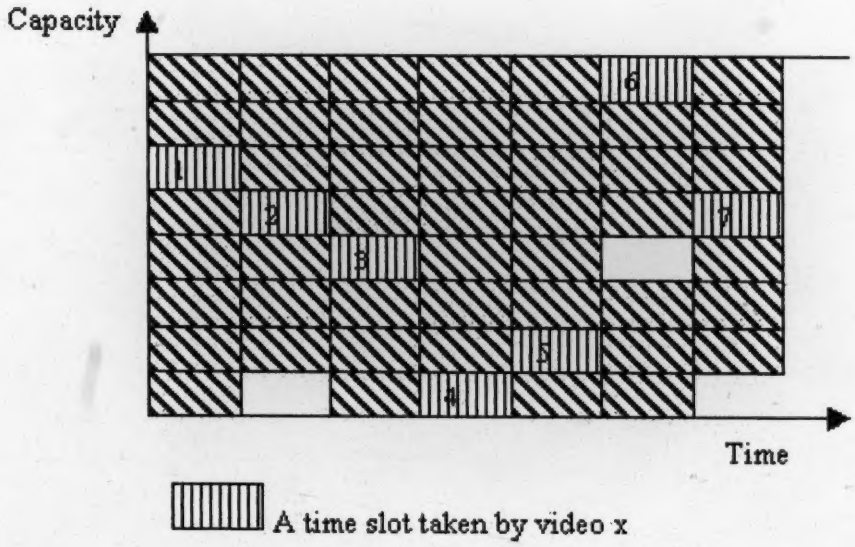

Figure 3.6: Continuous Time Slot for a Video

for ESF_S.

Algorithm ESF_S

```
Found=false; Number_of_Segment = d_trans / time_slot
For j=1 to Number_of_Segment do
    i=t_arrive
    while (i <= t_start) and not Found) do
        if  a cell in column i is available
            Found=true
        else
            i=i+1
    if Found
        t_start = t_start + time_slot
```

```
        Found=false
    else
        break
if Found
    accept the request
    insert segments into corresponding columns
else
     reject the request
```

LSF_S

LSF_S is similar to ESF_S. Each segment of a video has its own start time. Unlike in ESF_S where the server tries to allocate each segment as early as possible; in LSF_S, the server tries to allocate each segment as late as possible.

Here is the algorithm for LSF_S.

Algorithm LSF_S

```
Found=false; Number_of_Segment = d_trans / time_slot
For j=1 to Number_of_Segment do
  i=t_start
  while (i >= t_arrive) and not Found) do
      if  a cell in column i is available
          Found=true
      else
          i=i-1
  if Found
      t_start=t_start + time_slot
      Found=false
```

```
  else
      break
if Found
   accept the request
   insert segments into corresponding columns
else
  reject the request
```

3.5.3 Concentrated Scheme

In this scheme, the bandwidth is not divided into channels; the service provider uses the whole bandwidth to transmit a video.

Time slots are not used in this scheme. The reason is that if a video stream can only take part of a time slot, the remaining part of that slot will be wasted, since each stream starts from the beginning of each time slot. (See Figure 3.7)

We use a linked list to represent the bandwidth, and t_{arrive} *is the request arrival time,* t_{start} *is the anticipated start time,* d *is the duration of the requested video. Let* d_{trans} *be the duration of the transmission of this video. Then*

$$d_{trans} = d/bandwidth \tag{3.1}$$

The scheduled video is stored in a list. Each item in the linked list has properties: t_{arrive}, t_{start} *and* t_{end}.

$$t_{end} = t_{start} + d_{trans} \tag{3.2}$$

The t_{start} *here is the actual start time scheduled by the admission control server and* t_{end} *is the transmission finish time of the video. When a request*

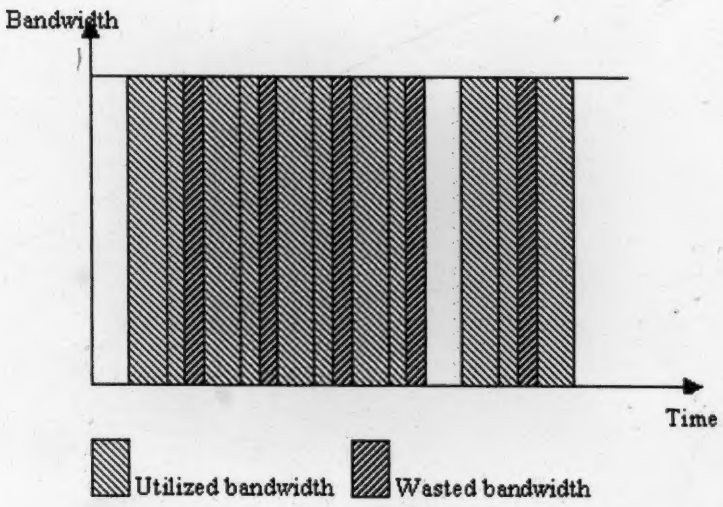

Figure 3.7: Time-slot in Concentrated Scheme

arrives, the admission control process searches the linked list to find a place to insert the new item. If the video can be fit in a gap between two scheduled videos or the t_{end} of the last video in the list is before the requested start time, the request will be accepted and the item will be inserted into the linked list by the scheduler; otherwise the request will be rejected.

Here is the algorithm for ESF_C and LSF_C respectively:

Algorithm ESF_C

```
Found=false; h=the head of the linked list
while (h <> the end of the linked list and not Found) do
   if (h.t_end <= request.t_start)
        Found=true
   else
```

```
        h=h.next
if Found
   accept the request
   insert "request" into the linked list
   this video is scheduled to start to transmit at
   max(request.t_arrive, h.t_end)
else
    reject the request
```

Algorithm LSF_C

```
Initialize a new item ''request" with properties:
                t_arrive, t_start and t_end
Found=false; h=the head of the link
Looking for a item whose start time is later than
request.t_start and whose previous item's t_start is
earlier than request.t_start
Let h points to this item
while (h <> the head of the linked list and not Found) do
  if (h.t_end <= request.t_start) and
     (request.t_end <= h.next.t_start)
     Found=true
  else
     h=h.previous
if Found
  accept the request
  insert ''request" into the linked list
  this video is scheduled to start to transmit
  at max(request.t_arrive, h.t_end)
```

```
else
  reject the request
```

The above algorithms only give the solutions for general situations. They do not include some special scenarios, e.g. the linked list is empty; the request must be inserted in front of the head or after the end of the linked list, etc. However, all of these scenarios are considered in our simulator.

3.6 Merging

A broadcast network environment is used in the video on-demand service, which means that the servers and clients are on a shared network media and multicast is easily supported on these networks. Multicast is a process that transmits data from a source to multiple destinations with one data stream, rather than with multiple streams. Multicast can greatly reduce the network traffic for bandwidth-hungry applications such as video on-demand.

As we know, hot videos have a high probability to be selected. The famous 80-20 theory tells us that 80% of requests are for the hottest 20% videos. Advance reservation can exploit the multicast capability and this frequency of repeated video requests by merging requests as long as the required start times are satisfied. This may save a lot of bandwidth.

The merging algorithm we used is: when an advance reservation for a video arrives, we look up the requested video in the schedule of ongoing videos, if this video is already scheduled to transmit no later than the start time of the new request, and it has not begun to transmit, then we merge the new request into that one.

3.7 Summary

We gave an overview of immediate reservation and advance reservation algorithms in this chapter. LSF with the extended transmission scheme is the traditional scheme in advance reservation systems. However it introduces fragmentation problems which will waste lot of bandwidth. To improve the bandwidth utilization we proposed a scheduling algorithm (ESF) and two transmission schemes (the segmented scheme and thte concentrated scheme). In ESF, the video will be transmitted as early as the bandwidth is available before the requested start time. Segmented scheme is the main contribution of this thesis, In the following chapter, we will give the simulation results and the comparison of these algorithms and schemes.

Chapter 4

Comparison of Proposed Schemes

4.1 Introduction

In this chapter we will describe our simulation model first, then the performance of different algorithms and schemes will be compared. For Earliest Start-time First (ESF) and Latest Start-time First (LSF), the performance of extended, segmented and concentrated schemes are compared respectively. We will also study the effect of immediate requests on our advance reservation system, and the cooperation of some other less bandwidth-consuming applications, e.g. voice applications with video applications.

4.2 Simulation Model

In our simulator, requests arrive at a video server and decisions are made to accept or reject them based on the availability of bandwidth. Requests are generated by the simulator according to a specific traffic pattern. The efficiency of the scheduling scheme is measured in term of the number of requests

being rejected and the bandwidth utilization. In this section the model of our simulation is given. We will also discuss two issues for analyzing the simulation results: transient removal and confidence intervals.

4.2.1 Parameters and Workload

In order to describe the experimental model, we first discuss details of the simulated workload.

We assume that the video requests are generated by a Poisson process which has an interval time that is exponentially distributed with a mean of 1 / λ. Some requests are immediate; others are in advance. The proportion of immediate requests is adjustable.

Based on video store rental patterns, we assume that accesses to movies in a video server will be highly localized, with a small number of "hot" movies receiving most of the accesses which follows the 80/20 law we presented before. We use Zipf's Law to characterize this locality. Zipf's Law states that the probability of choosing the ith most popular of N movies is:

$$f_i / \sum_{i=1}^{N} f_i \tag{4.1}$$

Where

$$f_i = 1/i \tag{4.2}$$

For example, using this distribution, the fifth most popular movie is requested one fifth as often as the most popular movie.

The notice intervals are uniformly distributed between 1 hour and 24 hours. At the server end, we assume the total bandwidth is equivalent to 84 MPEG-1

channels. We picked the number 84 to keep the rejection probability around 5 percent. Our video library has 10,000 videos. The duration of each video is given by a normal distribution with mean 90 minutes, deviation 20 minutes.

4.2.2 Performance Metrics

This section describes the metrics used in the performance analysis at the video distribution service system. The number of rejected requests and the bandwidth utilization form the basis for developing the performance metrics.

- ***Overall rejection probability:*** The overall rejection probability indicates the number of arriving requests per time unit at the video distribution service system that will be rejected. It includes the rejection for ReRA requests and ImRe requests.

- ***Class-based rejection probability:*** The requests are divided into ReRA and ImRe. Rejection probability for each class of reservation is defined as:

$$P_{Rej_ReRA} = \frac{Number_of_ReRA_Request_Rejected}{Total_Number_of_ReRA_Request} \tag{4.3}$$

$$P_{Rej_ImRe} = \frac{Number_of_ImRe_Request_Rejected}{Total_Number_of_ImRe_Request} \tag{4.4}$$

- ***Bandwidth utilization:*** The bandwidth utilization indicates the number of channels utilized per time unit. Suppose m requests are accepted. The overall bandwidth utilization is defined by:

$$B = \frac{\sum_{i=1}^{m} duration_of_video_i}{number_of_channels \times simulation_time} \tag{4.5}$$

- ***Capability to accept non-video requests:*** *Requests from other applications, e.g. voice and data, are introduced to the reservation system. Since the bandwidth requirement for these application is much lower than video application, they might be allocated in some gaps in the bandwidth where a video stream can not fit in. The capability to accept non-video requests is measure by how many non-video requests are accepted without increasing the rejection probability for videos.*

4.2.3 Simulation Methodology

Transient

Simulations are designed to predict the performance under stable conditions. During simulation, measurements should be taken after the system reaches a stable state. Therefore, the results of the initial part of the simulation should be dropped. This initial part is called the transient state. The end of the transient state must be identified to remove the transient. However it is almost impossible to find the exact end of the transient state. Therefore all transient removal methods are heuristic. Here is the method we used to remove the transient.

We repeat the simulation 8 times. Each replication consists of a complete run of the simulation with the same parameters but different seed values used in the random number generator. In each replication, 200,000 requests arrival are simulated.

Let x_{ij} denote the jth observation in the ith replication. The method consists of the following steps[19]:

1. *Get a mean trajectory by averaging across replications:*

$$\overline{x_j} = \frac{1}{m}\sum_{i=1}^{m} x_{ij}, \qquad j = 1, 2, ..., n \tag{4.6}$$

2. *Get the overall mean:*

$$\overline{\overline{x}} = \frac{1}{n}\sum_{j=1}^{n} \overline{x_j} \tag{4.7}$$

Set l=1 and proceed to the next step.

3. *Assume that the transient state is l long, delete the first l observations from the mean trajectory and get an overall mean from the remaining $n - l$ values:*

$$\overline{\overline{x_l}} = \frac{1}{n-l}\sum_{j=l+1}^{n} \overline{x_j} \tag{4.8}$$

4. *Compute the relative change in the overall mean:*

$$Relativechange = \frac{\overline{\overline{x_l}} - \overline{\overline{x}}}{\overline{\overline{x}}} \tag{4.9}$$

5. *Repeat step 3 and 4 by varying l from 1 to $n - 1$. After a certain value of l, the relative change will be stable. Then l is the transient interval.*

Every time a request arrives, we calculated the overall rejection probability which is considered as one observation. The number of observations in our experiment is the total number of requests arrive at the system which is 200,000 in this case. Since the average request arrival rate is around 60 requests per hour, the length of simulation is about 3000 hours. Different scheme or same scheme with different parameters will have different transient interval. We calculated the transient for each scenario by applying the algorithm above, and found that the maximum transient interval l is 19,000.

Confidence Interval

Every time we run a simulation, we may get different result. Sometimes the variance of the results might be very high. At this time, a mean of the results is not enough because a perfect mean can not be observed from a finite number of simulation runs. However, it will be acceptable if the mean is in between a pair of probabilistic bounds with a very high probability. Let c_1 *and* c_2 *as the lower bound and upper bound,* μ *as the mean,* $1-\alpha$ *as the high probability that the mean is in* (c_1, c_2)*:*

$$P\{c_1 \leq \mu \leq c_2\} = 1 - \alpha \tag{4.10}$$

The interval (c_1, c_2) *is called the confidence interval for the population mean,* α *is called the significant level,* $100(1-\alpha)$ *is called the confidence level, and* $1-\alpha$ *is called the confidence coefficient. Confidence level is generally presented as a percentage and is near 100%. In our experiment, 95% confidence intervals were found for the results. The formula used for 95% confidence interval (C.I.) is:*

$$95\%_C.I. = \mu \pm 2.36 * \frac{SD}{\sqrt{N}} \tag{4.11}$$

where SD is the standard deviation, N is the sample size.

4.3 Reservation Results

In this section we give the simulation results analysis for advance reservation and immediate reservation. We also explore the effect of merging requests and accommodating low bit rate transmission such as voice and data file.

A number of experiments have been run to compare these algorithms which are described in Sections 3.4 and 3.5. Of particular interest is the effect of

	58 Requests/Min			60 Requests/Min		
	Mean	C1	C2	Mean	C1	C2
Concentrated	0.014686	0.014292	0.01508	0.048577	0.04829	0.048864
Extended	0.021254	0.02086	0.021648	0.054794	0.05445	0.055143
Segmented	0.021261	0.020866	0.021656	0.054799	0.054458	0.05514
	62 Requests/Min			64 Requests/Min		
Concentrated	0.079134	0.078639	0.079629	0.108079	0.107603	0.108555
Extended	0.085208	0.084661	0.085755	0.114020	0.113697	0.114343
Segmented	0.085231	0.084683	0.085779	0.114054	0.113736	0.114372

Table 4.1: Confidence Interval: ESF

segmentation and merging on the probability of rejection and the bandwidth utilization.

To keep the rejection probability around 5%, 84 channels are used in our simulation. The advance reservation is made from 1 hour to 24 hours ahead of the start time.

4.3.1 Advance Reservation Results

In the first experiment, schemes using Earliest Start-time First (ESF) algorithm are considered, and all requests are placed in advance. Figure 4.1 shows the rejection probability for various load values from 58 requests per hour to 64 requests per hour. The "Concentrated" scheme accepts more requests than the other two. The rejection probabilities for "Extended" and "Segmented" are almost the same. The confidence interval is shown in Table 4.1.

The reason why the "Concentrated" scheme is better is shown in Figure 4.2 and Figure 4.3. In the "Concentrated"" case, we assume that the bandwidth before time t2 is allocated when a request comes at time t1 and the required start time is later than t2. Based on our algorithm, the bandwidth between t2 and t4 will be allocated to the request. If the next request comes at t3 and the

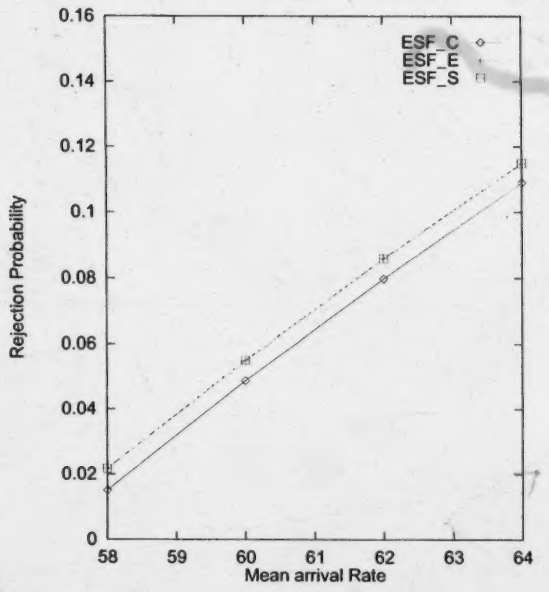

Figure 4.1: Earliest Start-time First: Advance Reservation

desired start time is later than t4, it will be scheduled to start from t4. From the figure we can see that no bandwidth is wasted. But in the "Extended" case, only one channel is utilized between t2 and t3; other channels are wasted.

In the second experiment the schemes using Latest Start-time First (LSF) algorithm are considered. Figure 4.4 shows the rejection probability for various load values from 58 requests per hour to 64 requests per hour. The "Concentrated" scheme still accepts more requests than the other two. But this time the "Segmented" scheme is much better than the "Extended" one. The "Extended" scheme needs a continuous bandwidth whose length is equal to the video length and only one logical channel can be used in each time interval. However the "Segmented" scheme does not have this restriction. For example, if a video must be transmitted for time t1, but bandwidth from t2 is full, and

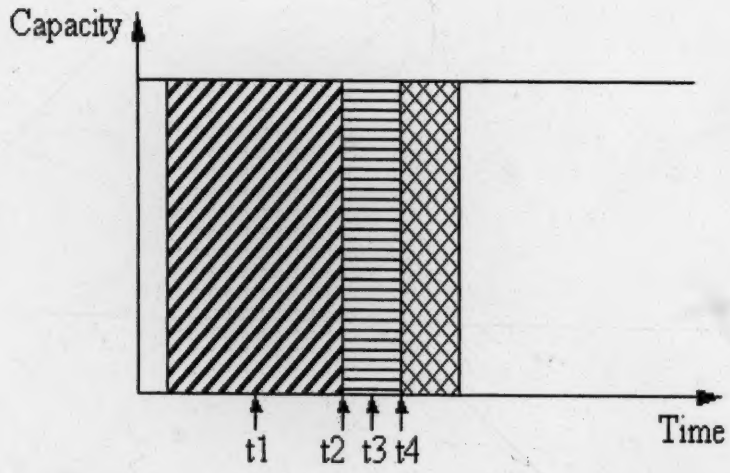

Figure 4.2: ESF-Concentrated: Bandwidth Utilization

$$t2 - t1 < video_length \tag{4.12}$$

then the request is rejected in the "Extended" scheme, but in the "Segmented" scheme, the first m segments are transmitted in time interval [t1,t2] continuously; the remaining segments can be transmitted in any other channels between t1 and t2.

In Figure 4.5 to 4.7, comparisons of the performance of the schemes in ESF and LSF are shown. All three schemes perform better in ESF than in LSF, and the "Extended" scheme even performs much better in ESF since LSF introduces more fragments in the bandwidth. Also, in the "Segmented" scheme, the performance of ESF and LSF becomes more similar at higher workload.

The reason that ESF performs better is: bandwidth is fully used by video streams. When the mean request arrival rate is high, the whole bandwidth is

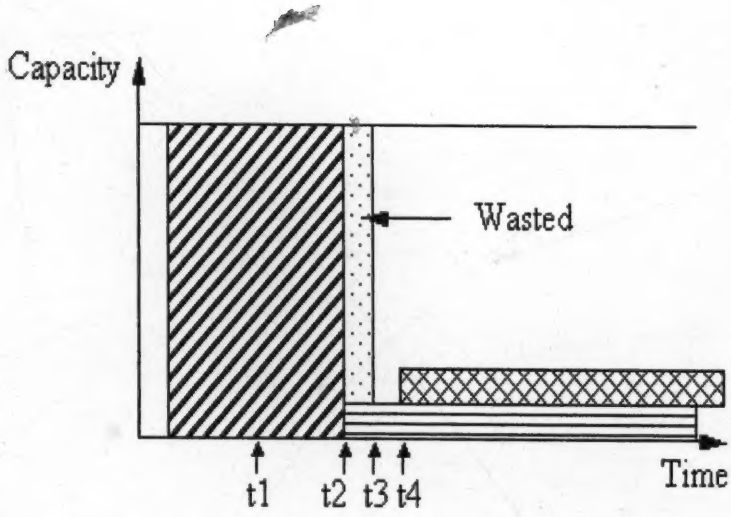

Figure 4.3: ESF-Extended: Bandwidth Utilization

utilized in the ESF scheme. But in LSF, the video is transmitted as late as possible, then the bandwidth between the arrival time and start time may not be usable. That is, when a request R_i arrives at time t1, even if bandwidth is available at t1, we will still wait until its start time. Only when no bandwidth is available at the desired start time , may the transmission of a stream begin earlier. Let's assume the next request R_{i+1} arrives at t2, and R_i will be transmitted after t2 or R_i is allocated in (t1, t2) but it can not take the whole bandwidth in (t1, t2), then the available bandwidth in (t1, t2) is wasted in "Concentrated" and "Extended" schemes since the bandwidth before t2 can not be allocate to R_{i+1} in these 2 schemes. Although the "Segmented" scheme may make use of the bandwidth in (t1, t2), a segment will not be transmitted in (t1, t2) unless the workload is too high. This also explains why ESF and LSF perform very similarly at high workload in the "Segmented" scheme. Because the average notice interval is 12 hours, the segments will be allocated 12 hours after t2 if possible and it is rarely allocated in (t1, t2).

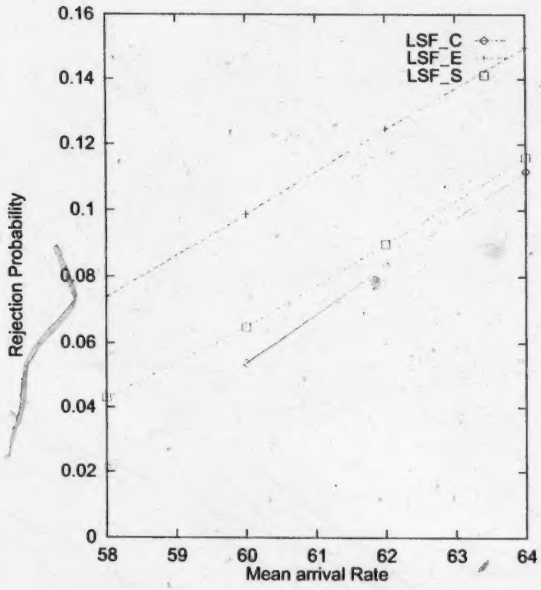

Figure 4.4: Latest Start-time First: Advance Reservation

4.3.2 Mixed Advance and Immediate Reservation

In the next experiment, the effect of ImRe requests on the system is studied. In these experiments, the ImRe request is rejected if no bandwidth is available immediately. ESF is used for the advance reservation case. Figure 4.8 shows that when ImRe requests make up 10 percent of the total requests, the "Extended" scheme performs the best when the work load is not very high. Actually we are only interested in the workload which gives us 5 to 10 percent rejection. If the rejection probability is more than 10 percent, subscribers will begin to complain, and it is time for the ISP to increase their bandwidth. In ESF, a video takes the whole bandwidth in "Concentrated" for a period of time or several channels in "Segmented" in some time intervals. It only takes one channel in "Extended". If the next ImRe request arrives at those time intervals, it must be rejected in the "Concentrated" scheme and has more chance

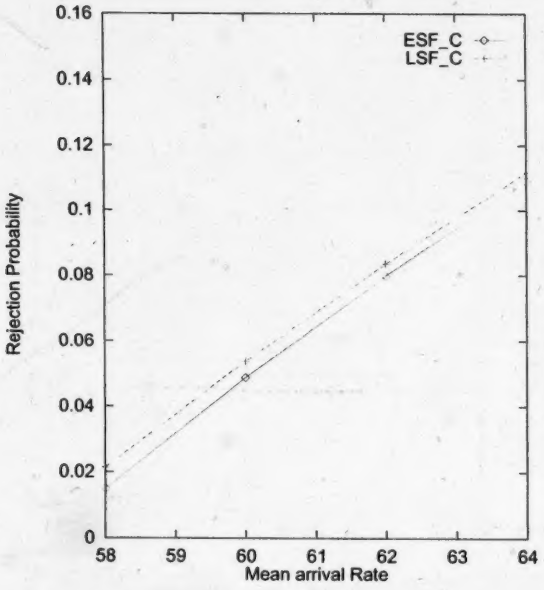

Figure 4.5: Concentrated Scheme: ESF vs. LSF, Advance Reservation

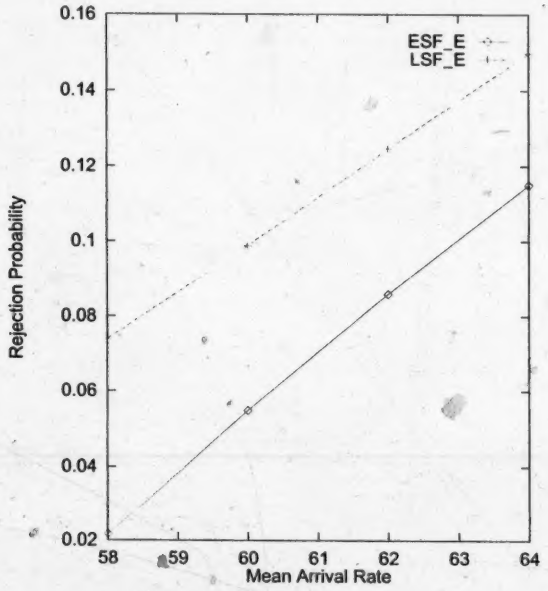

Figure 4.6: Extended Scheme: ESF vs. LSF, Advance Reservation

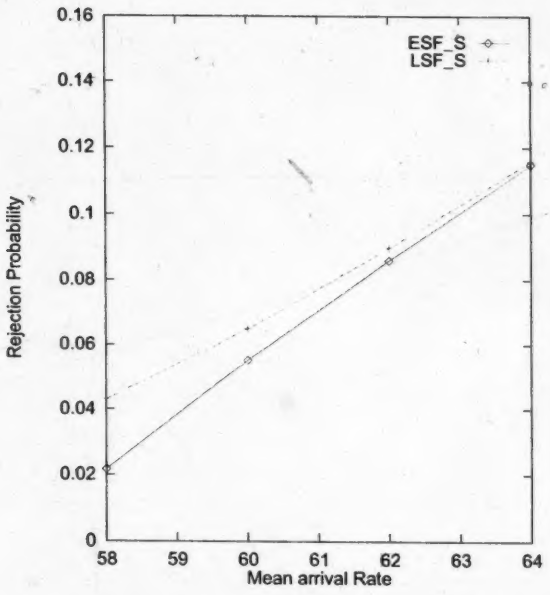

Figure 4.7: Segmented Scheme: ESF vs. LSF, Advance Reservation

	58 Requests/Min		60 Requests/Min		62 Requests/Min		64 Requests/Min	
	ReRA	ImRe	ReRA	ImRe	ReRA	ImRe	ReRA	ImRe
Concentrated	0	77.89%	0	86.74%	0	94.71%	0.88%	99.99%
Extended	0	41.10%	0	63.99%	0	88.26%	1.46%	99.95%
Segmented	0	64.56%	0	79.63%	0	93.57%	1.46%	99.99%

Table 4.2: Rejection Proportion, ReRA vs. ImRe (ESF)

to be rejected in the "Segmented" scheme. Table 4.2 shows the percentage of rejected ReRA request and percentage of rejected ImRe request. We can see almost all of the rejected requests are ImRe requests.

Figure 4.9 shows that when ImRe requests make up 10 percent of the total requests and LSF is used, the "Segmented" scheme performs the best when load is not very high. From Table 4.3 we can see the "Concentrated" does not do well with ImRe request. We have already known that the "Segmented" scheme performs better than "Extended" scheme in LSF, and "Concentrated"

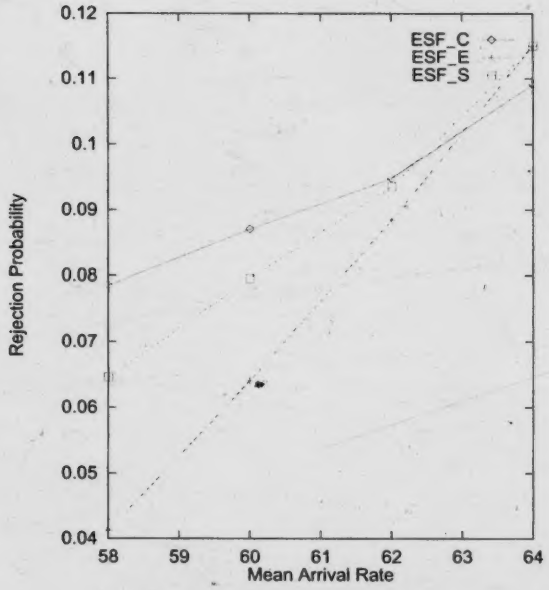

Figure 4.8: Earliest Start-time First: 10% ImRe

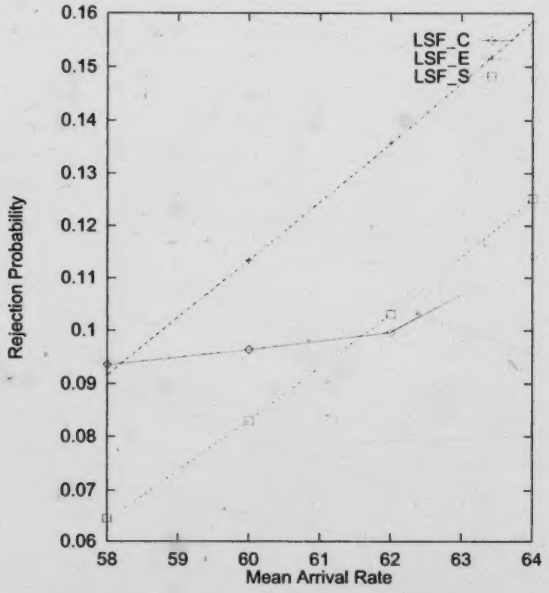

Figure 4.9: Latest Start-time First: 10% ImRe

	58 Requests/Min		60 Requests/Min		62 Requests/Min		64 Requests/Min	
	ReRA	ImRe	ReRA	ImRe	ReRA	ImRe	ReRA	ImRe
Concentrated	0	100%	0	100%	0	100%	1.62%	99.99%
Extended	2.14%	72.13%	3.42%	82.33%	5.13%	89.56%%	7.09%	94.66%
Segmented	0.85%	56.61%	1.60%	68.38%	2.73%	78.87%%	4.19%	87.16%

Table 4.3: Rejection Proportion, ReRA vs. ImRe (LSF)

performs the best there. But when ImRe is introduced, from Table 4.3 we can see almost all the ImRe requests are rejected in the "Concentrated" scheme, but much fewer ImRe requests are rejected in the "Segmented" and "Extended" schemes. So "Segmented" keeps doing better than the "Extended" scheme, and the rejection probability of "Concentrated" increases sharply. But when the workload goes up, the "Segmented" and "Extended" schemes will reject as many ImRe requests as "Concentrated" scheme does. Then "Concentrated" goes back to being the best among these three schemes. That is why "Concentrated" has a different shape in the diagram.

In the following, ESF and LSF are directly compared when there are 10 percent ImRe requests. Figures 4.10, 4.11 and 4.12 show that when ImRe requests take 10 percent of all requests, all three schemes perform better in ESF although their rejection probabilities are higher than advance reservations respectively (see Figure 4.13). These three figures have the similar shape as the Figure 4.5 to 4.7, and ESF gets better performance due to the same reason. When ESF performs better than LSF, LSF must leave more segments here and there unused. But we can also see that the difference between ESF and LSF on the "Segmented" scheme is smaller than the "Extended". This is because, although LSF introduces more fragments, the "Segmented" scheme can make use of those fragments to save bandwidth.

Now, let's have a look at the change of rejection probability when the

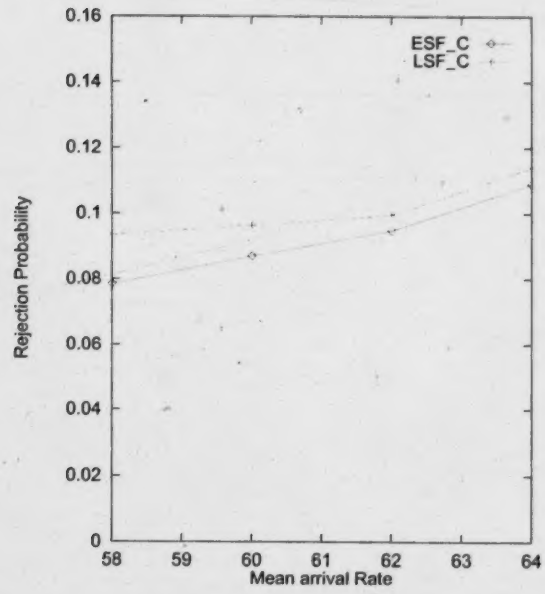

Figure 4.10: Concentrated Scheme: ESF vs. LSF, 10% ImRe

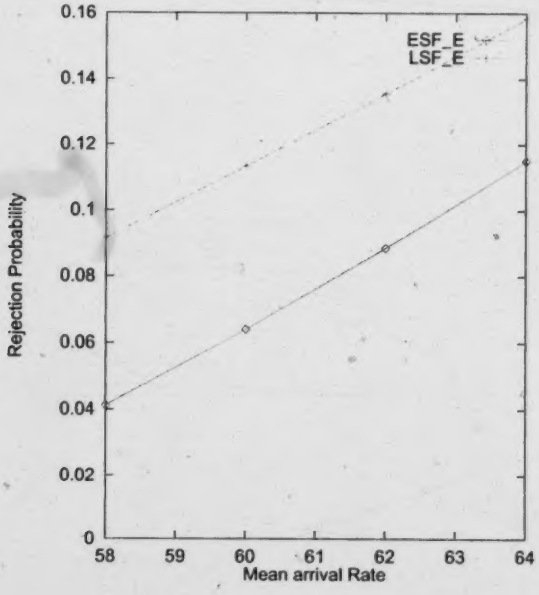

Figure 4.11: Extended Scheme: ESF vs. LSF, 10% ImRe

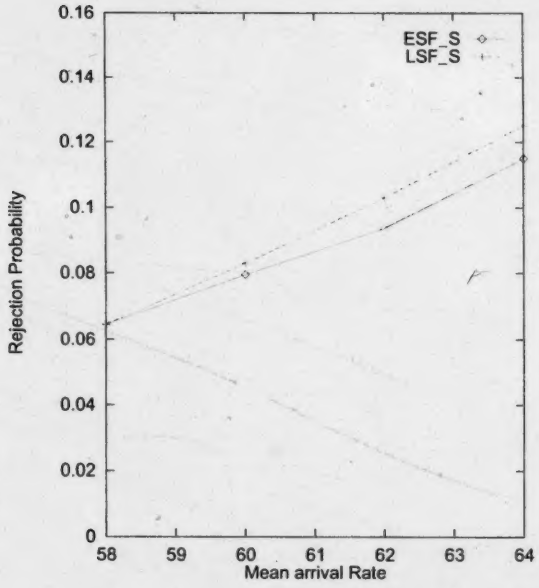

Figure 4.12: Segmented Scheme: ESF vs. LSF, 10% ImRe

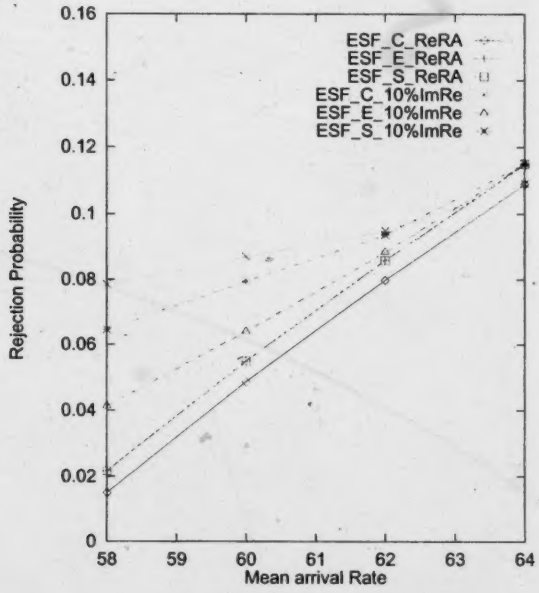

Figure 4.13: Impact of 10% ImRe

proportion of ImRe requests is varied from 0% to 100% and the mean arrival rate is kept at 60.

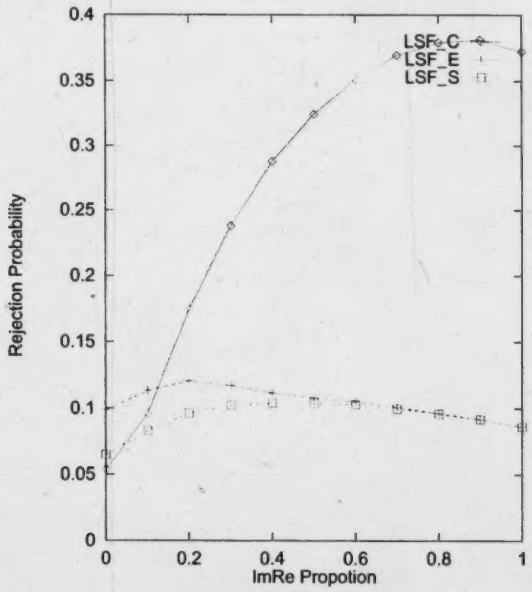

Figure 4.14: Latest Start-time First: ReRA and ImRe

From Figure 4.14 we find that the "Concentrated" scheme can not deal with ImRe requests very well in LSF, because it always takes the whole bandwidth during a certain time slot. The ImRe requests that arrive at that time slot will be blocked. When there are mixed ReRA and ImRe requests, the videos reserved by advanced reservation will be put in the pipe some time away from the requests arrival time. But the videos reserved by ImRe will be put in the pipe immediately. We can consider that the videos for advance reservation are allocated at the rear of the pipe while the videos for ImRe requests are allocated at the front of the pipe and the gap between them is getting smaller. Eventually no video can be allocated to this gap, more rejections will occur. But if 100 percent requests are ImRe, the video will be piled up in the pipe, just

like what the ESF does. The work load we used is suitable for the "Extended" and "Segmented" schemes, but it is too high for the "Concentrated" scheme, so the system rejects a lot in the "Concentrated" scheme.

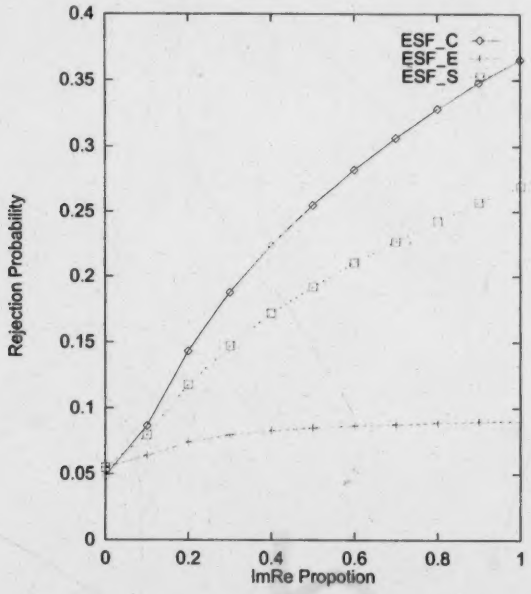

Figure 4.15: Earliest Start-time First: ReRA and ImRe

Figure 4.15 shows that when the proportion of on demand requests increases, the rejection probability for the "Segmented" scheme goes up quickly in ESF. The segments of a video are piled up and packed when the earliest bandwidth is available, its shape is similar to the "Concentrated" scheme. So its performance is getting close to "Concentrated", more ImRe requests will be rejected.

Delay for ImRe Request

In the next experiment, we do not reject the ImRe request immediately if no bandwidth is available. We try to allocate the bandwidth by delaying the

request for a certain period of time. Since the user will not like waiting, we will not delay his request too long and will set a threshold for the delay. If a request can not be satisfied within the threshold, it will be rejected. Figure 4.16 shows that the rejection probability decreases when we increase the threshold in LSF. The delay we used in our experiment varies from 60 seconds to 1000 seconds. But we do not see a significant decrease when we increase the threshold. The highest threshold is only 1000 seconds here, it is the same as an advance reservation with 1000 seconds notice interval. The other 90% requests are advance reservation with mean notice of 12 hours which still give a big advantage over the delayed ImRe requests. So the rejection probability does not decrease too much, and delayed ImRe requests are still getting rejected.

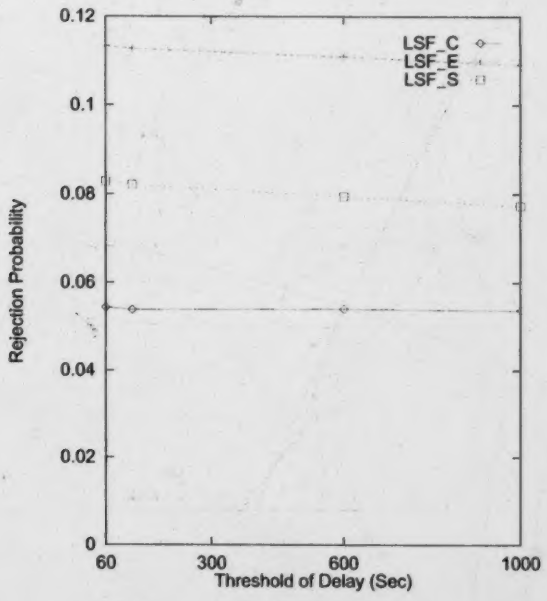

Figure 4.16: Latest Start-time First: 90% Advance Reservation, Delay

To see how many on-demand requests are denied, we set the threshold to a

large value, then we find that many more on-demand requests are rejected than advance requests. Actually, ReRA has higher priority than ImRe because it arrives earlier. If a channel is booked by an advance reservation, the channel can not be allocated to an immediate request. So a client who uses ReRA will get much more chance to get bandwidth than the traditional reservation protocols. Table 4.4 shows the rejection probability for ImRe requests when ImRe requests makeup 10% of total requests and the mean request arrival rate is 60. From the table we can see that, even though ImRe requests take only 10% of total requests, most of them are blocked. Since they did not reserve the bandwidth in advance, the service provider can not guarantee that they get it.

	Total	ReRA	ImRe
Concentrated	9.64%	0	96.43 %
Extended	11.34%	3.42%	82.33%
Segmented	8.30%	1.60%	68.38%

Table 4.4: Rejection Proportion for ImRe Requests (No delay)

Table 4.5 shows the results when ImRe requests makeup 10% of the total requests and the threshold for delay is 600 seconds. That means if desired bandwidth will become available in 600 seconds, the request will be accepted. Here we used the LSF algorithm, because from the previous section, we can see that ImRe does not work well in ESF. In our experiment, no matter whether the request is delayed or not, most requests being denied are ImRe ones in LSF. And in ESF, when a request is not delayed, all the ImRe requests are rejected.

In this section we see that the "Segmented" scheme performs the best in LSF, and "Extended" scheme wins in ESF with mixed ReRA and ImRe traffic. But the "Concentrated" scheme does not work well in neither ESF nor LSF when ImRe traffic exists.

	Total	ReRA	ImRe
Concentrated	5.40%	5.36%	5.76%
Extended	11.09%	3.47%	79.63%
Segmented	7.95%	1.69%	64.32%

Table 4.5: Rejection Proportion for ImRe Requests (Threshold=600)

4.3.3 Merging

Hot videos have a high probability of being selected. The famous 80-20 theory tells us that 80% requests are for the hottest 20% videos. So if we can merge the requests for the same video and then multicast it to multiple clients, it will save much bandwidth. The merging algorithm we used was described in Section 3.6. In Table 4.6 we use C_M to denote the "Concentrated" scheme where merging is being used, C_NM to denote the "Concentrated" scheme where merging is not being used, and so on.

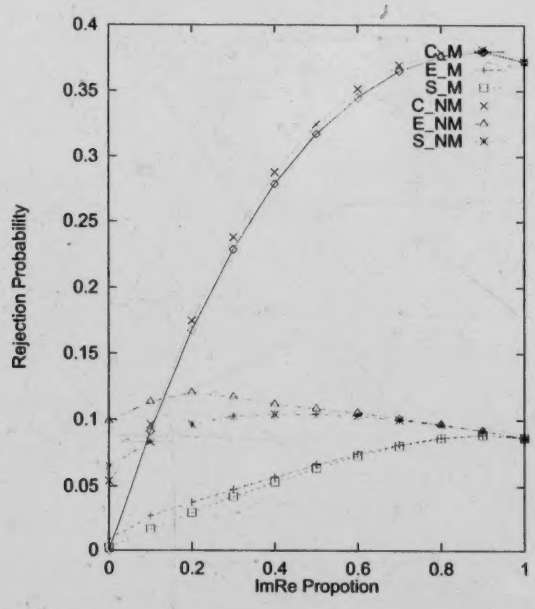

Figure 4.17: Latest Start-time First: Merging, No Delay

Figure 4.17 and Table 4.6 show that the rejection probability drops significantly in LSF when merging is used and no ImRe requests are presented. From the last section we can see that almost all rejected requests are ImRe requests when the "Concentrated" scheme is used, and we also know that ImRe requests can not be merged with ImRe requests, but it may be merged with ReRA requests rarely. So when the ImRe is allowed in our system and the ImRe requests are not delayed, the "Concentrated" scheme performs almost the same with merging or without merging.

We can also see that merging still improves the performance quite a bit in the other two schemes when the percentage of ImRe requests is not very high. When the percentage of ImRe requests goes up, the advantage of merging will disappear because merging works only for advance reservation unless an ImRe request by any chance arrives at the same time as a pre-scheduled reservation.

LSF	0	0.1	0.3	0.5	0.7	0.9
C_M	0	0.0912	0.229	0.3171	0.3652	0.3802
C_NM	0.0538	0.0964	0.238	0.3245	0.3698	0.3813
E_M	0.0085	0.0275	0.0476	0.0661	0.0812	0.0886
E_NM	0.0988	0.1134	0.1171	0.1083	0.101	0.092
S_M	0.0021	0.0172	0.0423	0.0638	0.0803	0.0885
S_NM	0.0648	0.0831	0.1026	0.1041	0.0999	0.0919

Table 4.6: Merging vs. Non-Merging (LSF, NO Delay, Mean Request=60)

When merging is used in ESF, the rejection probability drops sharply in all three schemes when no ImRe requests are present. However, from the last section we found that all ImRe requests will be rejected if we do not delay them. So merging does not work in ESF when ImRe reservation exists. Figure 4.18 and Table 4.7 compare the rejection probability in different schemes with merging or without merging for ESF.

ESF	0	0.1	0.3	0.5	0.7	0.9
C_M	0	0.083011	0.186719	0.254382	0.305771	0.347079
C_NM	0.048395	0.086376	0.187661	0.254914	0.306249	0.347675
E_M	0	0.055961	0.077884	0.084081	0.087051	0.089094
E_NM	0.054725	0.063759	0.079326	0.084874	0.087597	0.08964
S_M	0	0.074801	0.146224	0.191436	0.226541	0.256536
S_NM	0.054743	0.079341	0.147295	0.191966	0.227224	0.257243

Table 4.7: Merging vs. Non-Merging (ESF, NO Delay, Mean Request=60)

Bandwidth Utilization

The bandwidth utilization is a very important issue in our system. An ideal application should have low rejection and low bandwidth utilization. That means for a given mean request arrival, when the same percentage of requests are rejected, the scheme which utilizes less bandwidth is better. This saves bandwidth for other applications. However there are tradeoffs, if more requests are rejected, there usually will be more spare space in the bandwidth.

When we ran our simulation with merging, we also measured bandwidth utilization with different proportions of ImRe request. When we compare Figure 4.19 with Figure 4.17 (Rejection probability for LSF when merging is used) and Fugure 4.20 with Figure 4.18 (Rejection probability for ESF when merging is used), we can see when the rejection probability is higher, the bandwidth utilization is lower. But in some circumstances, which we are very interested in, for example, in LSF (Figure 4.19 and 4.17), when the proportion of ImRe requests is below 10% and merging is used, the "Extended" and "Segmented" scheme accept more requests than the "Concentrated" scheme does, and they also use less bandwidth. The reason as we mentioned before is that it is easier for the "Extended" and "Segmented" scheme to merge multiple requests.

We are not surprised to see that the "Concentrated" scheme has a very different shape in Figure 4.19. Comparing it with Figure 4.17 which shows its the rejection probability, we can see that the rejection probability for the "Concentrated" scheme goes up dramatically with the increase of the proportion of ImRe reservation, and in the meantime, the bandwidth utilization drops.

Another interesting result is in ESF, also when the proportion of ImRe requests is below 10%, the bandwidth utilization goes up with the rejection probability in all three schemes. This is more evidence that merging is not doing well in ESF when ImRe requests are present. Because a video is scheduled and transmitted as early as possible, later-coming ImRe request for the same video may not be merged with it.

From these above, we find that merging really improves the performance a lot. It not only decreases the rejection probability (See Figures 4.17 and 4.18) but also saves a lot of bandwidth (See Figures 4.22 and 4.21).

4.3.4 Voice and Data Requests

From the results in previous sections we find that there is still room to improve the performance in our system. Since the bandwidth is not fully loaded yet, why not use it to transmit more data? However we can not put more videos into it, because as we have seen that would increase the rejection probability sharply. What we can do is put some voice or data transfers into the bandwidth, because they take much less bandwidth than videos.

In our simulation, we assume the voice or data file transmission takes 1/30 of the bandwidth compare to video. The voice or data file requests are generated by a Poisson process as well and these requests are all ImRe requests. The scheme used for voice is "Extended" because for a real time

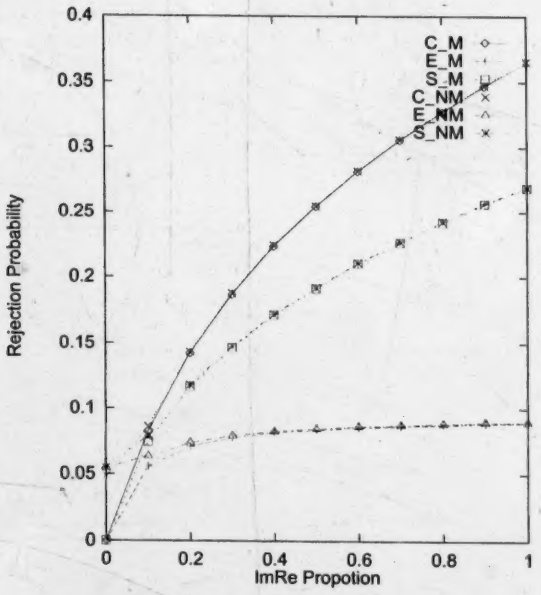

Figure 4.18: Earliest Start-time First: Merging, No Delay

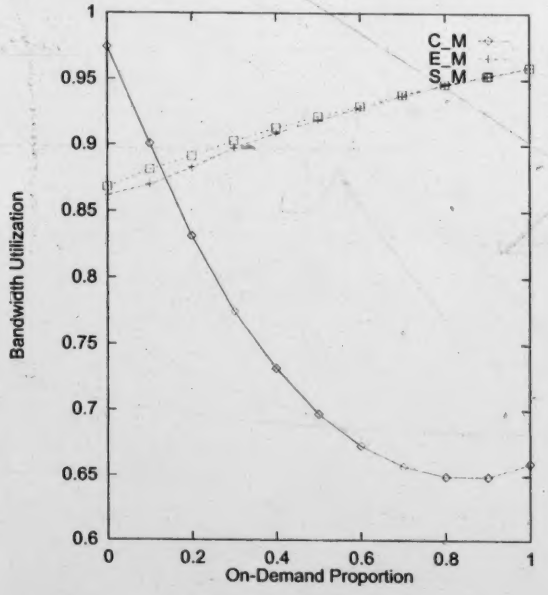

Figure 4.19: Latest Start-time First: Bandwidth Utilization

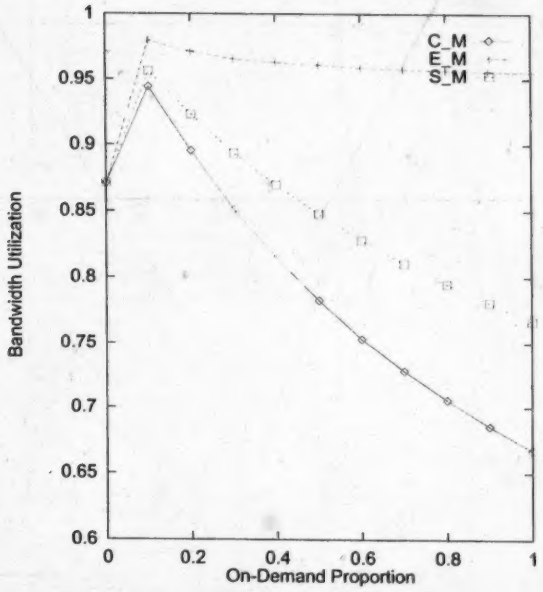

Figure 4.20: Earliest Start-time First: Bandwidth Utilization

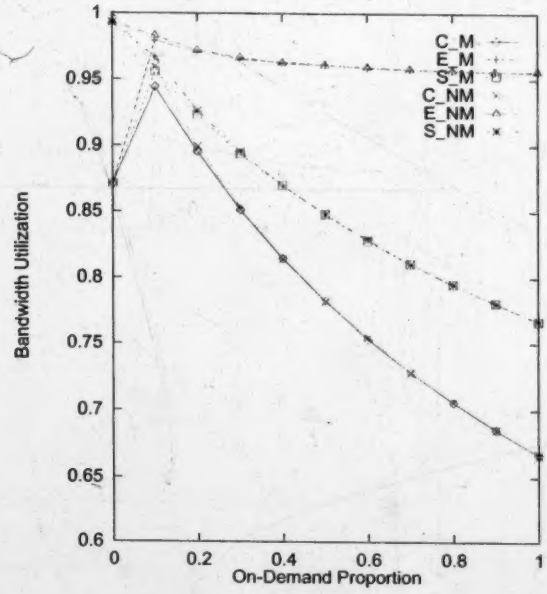

Figure 4.21: ESF Bandwidth Utilization Comparison

voice communication we cannot pre-fetch or delay the data or segment it. The mean video arrival rate is fixed at 60 per hour, but the voice arrival rate varies from 10 to 40 per hour. If the mean voice arrival rate reaches 40 per hour, our system accepts 60,000 to 70,000 voice or data transfer requests in the "Segmented" and "Extended" scheme when the simulation gets into its steady state. As can be seen in Figure 4.23, the rejection probability for video requests is very stable. However, the system cannot accept as many voice requests in the "Concentrated" scheme as in the other two schemes because a continuously extended voice stream cannot be fitted in a very thin gap in the "Concentrated" scheme.

4.4 Summary

The performance evaluation started with addressing the simulation model. The bandwidth is shared between advance reservation and on-demand reservation, and also between video and voice. We studied the impact of "LSF" and "ESF" scheduling algorithm and "Concentrated', "Extended" and "Segmented" schemes on advance reservation. Merging and multicast are used in our simulator to improve the performance. The performance can be expressed in term of rejection probability and total bandwidth utilization.

Actually, the "Concentrated" scheduling will not be used in a real system since it requires high bandwidth at he user end, and the ESF scheme has high requirements for the end user. So the LSF with "Extended" or "Segmented" scheme might be used by the ISP. However the "Segmented" scheme gives better performance than the currently used "Extended" scheme.

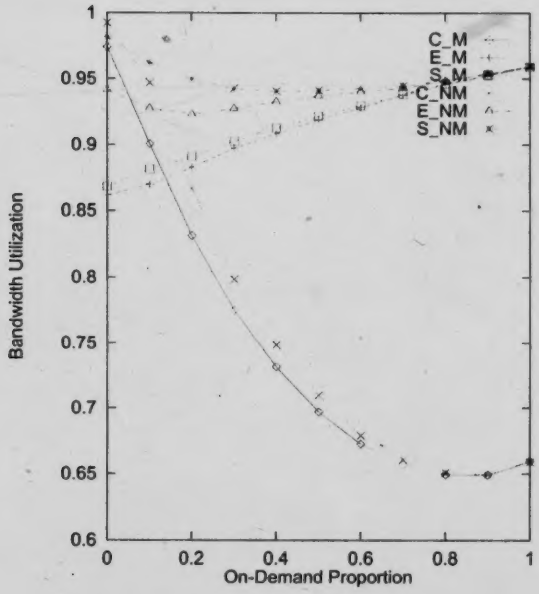

Figure 4.22: LSF Bandwidth Utilization Comparison

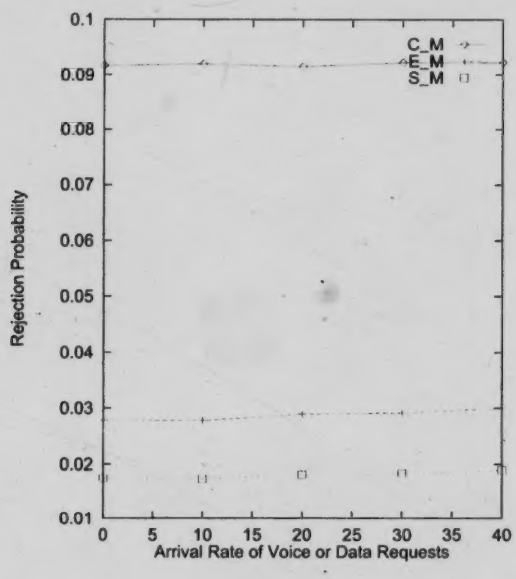

Figure 4.23: Rejection probability for Video when Voice and Data Requests are Added

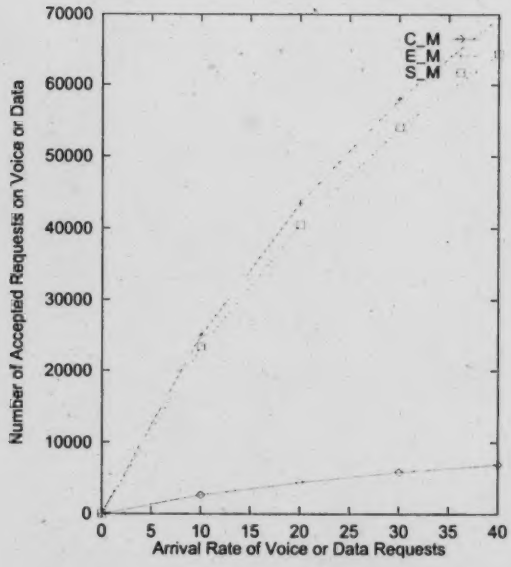

Figure 4.24: Number of Voice Requests Accepted

Chapter 5

Conclusions and Future Work

5.1 Conclusion

We expect to receive many services directly to the home in the upcoming years through the cable company or telephone network. Some of them are non-interactive services, like reserving a video in advance. These applications can give the service provider a notice of how much bandwidth they need and how long they need it. Then advance reservation is very suitable for them.

Two scheduling algorithms "Earliest Start-time First(ESF)" and "Latest Start-time First(LSF)" and three transmission schemes are proposed in this paper. A performance study has been presented for a video distribution system on the Internet. We also consider the effect of adding a voice and data service to the simulated system.

The performance study indicates that if no merging is used, all the three schemes ("Concentrated", "Extended" and "Segmented") perform better in ESF than in LSF. To make it clear, we use the tables below (Table 5.1 and Table 5.2) to give a comparison of their performance (in terms of rejection probability). The numbers in the tables indicate the order of the schemes,

number "1" is the best one. If two schemes have the same number, then their performance is almost the same.

		Concentrated	Extended	Segmented
ReRA	ESF	1	2	2
	LSF	1	3	2

Table 5.1: Performance Comparison: ReRA, Non-Merging

		Concentrated	Extended	Segmented
ReRA with ImRe, No Delay	ESF	3	1	2
	LSF	3	2	1
ReRA with ImRe, Delay	ESF	1	2	2
	LSF	1	3	2

Table 5.2: Performance Comparison: ReRA and ImRe, Non-Merging

From our study we found that advance reservation improves the performance significantly to handle more clients by merging requests. For those customers who request ImRe video services, if they can tolerate a limited time of waiting, the system can improve its performance even more by delaying some requests. The ImRe requests have less opportunity to get through, because they do not give the system enough time to schedule them. In the meantime, it is hard to merge ImRe requests with other requests, therefore ImRe requests are more sensitive to the load in the system.

The tables below(Table 5.3 and Table 5.4) show the performance comparison when requests are merged.

		Concentrated	Extended	Segmented
Pure ReRA	ESF	0% Rejection	0% Rejection	0% Rejection
	LSF	1	3	2

Table 5.3: Performance Comparison: ReRA, Merging

To utilize the bandwidth better, some interactive traffic is allowed in our

		Concentrated	Extended	Segmented
ReRA with ImRe, No Delay	ESF	3	1	2
	LSF	3	2	1
ReRA with ImRe, Delay	ESF	1	3	2
	LSF	1	3	2

Table 5.4: Performance Comparison: ReRA and ImRe, Merging

system, e.g. voice and data transfer. Then a lot of low bandwidth-consuming requests are accepted without affecting the video traffic.

In a realistic video distribution system, the "Concentrated" scheme would be rarely used, because the requirement for end users may be too high. The customers may not like the ESF although it gives the service provider much flexibility to schedule its transmission, because it requires the customer to buffer the video for a long time before his preferred viewing time, which will impact the customer's other download activities because of the preoccupied buffer. So what are more interesting are the "Extended" and "Segmented" schemes in LSF. It is easy to see from our simulation that advance reservation with the "Segmented" scheme and merging improves the performance of non-interactive services significantly, e.g. video distribution.

5.2 Future Work

Some issues remain for future work. We have used a simple model for the advance reservations. An advanced reservation can not be cancelled or revised in our simulator. In a real system, a user might want to change the start time of the reservation or even the video itself. So this feature might be added in the next version of simulator.

Another continuation would be to improve the merging algorithm. In the

current simulator, the later coming requests might be merged with previously scheduled transmission. But actually the former scheduled video also could be merged with a new request. Because the later arrival may have earlier start time, we can reschedule its start time to merge with the new request.

Bibliography

[1] *Janelle J. Harms, Video Distribution to the Home Using Advance Reservation. Proceedings of the 22nd IEEE Conference on Local Computer Networks (LCN '97)*

[2] *D. Ferrari, A. Gupta, and G. Ventre. Distributed advance reservation of real-time connections. In Proceedings of NOSSDAV'95, Durham, NC, April 1995.*

[3] *W. Reinhardt. Advance reservation of network resources for multimedia applications. In Proceedings of 2nd Intl. Workshop on Advanced Teleservices and High-Speed Communication Architectures (IWACA'94), Heidelberg, Germany, September 1994.*

[4] *"Interactive TV Trials", Sam's Telecom Casbah. http://www.teleport.com/ samc/cable4.html*

[5] *Miranda Ko and Irene Koo. An overview of interactive video on demand system. The University of British Columbia, December 1996.*

[6] *Introduction to DWDM Technology, Cisco Systems, Inc.*

[7] *SearchSystemsManagement.com Definitions*

[8] *http://www.cache.net/network/bandwidth_terms*

[9] *An Architecture for Differentiated Services, Request for Comments: 2475*

[10] *W. Reinhardt. Advance resource reservation and its impact on reservation protocols. In Proceedings Broadband Islands'95, Dublin, Ireland, September 1995.*

[11] *O. Schelen and S. Pink. An agent based architecture for advance reservations. In Proceedings of 22nd Annual Conf. Local Comput. Netw. (LCN'97), Minneapolis, MN, November 1997.*

[12] *O. Schelen and S. Pink. Sharing resources through advance reservations agents. In Proceedings of IWQoS'97, New York, NY, May 1997.*

[13] A. Schill, F. Breiter, and S. Kuhn. Design and evaluation of an advance resource reservation protocol on top of RSVP. InPro- ceedings of IFIP Broadband'98, Stuttgart, Germany, April 1998.

[14] O. Schelen, A. Nilsson, J. Norrgard, S. Pink: Performance of QoS Agents for Provisioning Network Resources. In Proceedings of IFIP Seventh International Workshop on Quality of Service (IWQoS'99), London, UK, June 1999.

[15] L.Zhang et al,RSVP:A new resource reservation protocol,IEEE Network ,vol.7 Sept.1993.

[16] IETF RSVP Working Group, http://www.ietf.org/html.charters/rsvp-charter.html

[17] A. Schill, S. Kuhn, and F. Breiter. Resource reservation in advance in heterogeneous networks with partial ATM infrastructures. In Proceedings of INFOCOM'97, pages 612 619, Kobe, Japan, April 1997.

[18] G. K. Zipf. Human Behaviour and the Principles of Least Effort. Addison-Wesley, Cambridge, MA, 1949.

[19] Raj Jain, The Art of Computer Systems Performance Analysis, Wylie, 1991.

[20] D. Wischik and A. Greenberg. Admission control for booking ahead shared resources. In Proceedings of INFOCOM'98, San Francisco, CA, April 1998.

[21] M. Degermark, T. Khler, S. Pink, O. Scheln: "Advance Reservation for Predicted Service", Fifth International Workshop on Network and Operating System Support for Digital Audio and Video, Durham, NH, USA, April 19-21, 1995.

[22] L.C. Wolf, L. Delgrossi, R. Steinmetz, S. Schaller, H. Wittig: "Issues of Reserving Resources in Advance", Fifth International Workshop on Network and Operating System Support for Digital Audio and Video, Durham, NH, USA, April 19-21, 1995.

[23] Lars C. Wolf, Ralf Steinmetz. Concepts for Resource Reservation in Advance. Special Issue of The Journal of Multimedia Tools and Applications on The State of The Art in Multimedia May 1997 (Vol. 4, No. 3)

[24] J. J. Harms and J. W. Wong. Performance modeling of a channel reservation service. Computer Net. ISDN Sys., 27:1487-1497, 1995.

[25] S. Berson, R. Lindell, and R. Braden. An Architecture for Advance Reservations in the Internet, July 1998. USC Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292. 16

[26] D. Wischik and A. Greenberg, Admission control for booking ahead shared resources, Proceedings IEEE Infocom '98, 1998.

[27] L. Gao and D. Towsley, "Supplying instantaneous video-on-demand services using controlled multicast," in Proc. IEEE International Conference on Multimedia Computing and Systems, 1999.

[28] S. Sen, L. Gao, J. Rexford, and D. Towsley, "Optimal Patching Schemes for Efficient Multimedia Streaming", Proc. 9 th Int'l Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'99), Basking Ridge, NJ, June 1999.

[29] K. Hua, Y. Cai, and S. Sheu, "Patching: A multicast technique for true video-on-demand services," in Proc. ACM Multimedia, September 1998.

[30] A.Gupta: Advance reservations in real-time communication services. Sun Microsystems Laboratory.

[31] A.Dan, D. Sitaram and P. Shahabuddin: Dynamic Batching Policies for an On-Demand Video Server. ACM Multimedia Systems, Vol 4, No 3, pp. 112-121, June 1996.

[32] C.C. Aggarwal and J.L. Wolf and P.S. Yu. "On Optimal Batching Policies for Video-onDemand Storage Server". Proc. of the IEEE Int'l Conf. on Multimedia Systems. June 1996.

[33] Y. H. Chang et al., "An Open-Systems Approach to Video on Demand," IEEE Communications, Vol. 32, No. 5, May, 1994, pp. 68-80.

[34] D. Deloddere, W. Verbiest, H. Verhille, "Interactive Video On Demand," IEEE Communications Magazine, May 1994, pp. 82-88

[35] Coffman and Odlyzko, The size and growth rate of the Internet, October 1998; Internet growth, forthcoming 2001.

[36] G.Bell. Ultracomputers: A teraflop before its time. Communications of the ACM, 35(8):26-47,1992

[37] http://about.reuters.com/newsreleases/art_25-6-2001_id639.asp

[38] http://www.anderson.ucla.edu/zone/clubs/htba/digi1129.ppt

[39] Harindra Rajapakshe, Derek Paul Quek, Video on Demand, SURPRISE 95 Journal, vol 4

[40] MPEG-4 Overview & Profiles, MPEG home page (www.cselt.it/mpeg).